

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

**Entwicklung einer optimalen
Migrationsstrategie für ein hierarchisches
Datenmanagement System**

Carolin Schmitz

FZJ-ZAM-IB-2004-07

Juli 2004

(letzte Änderung: 28.07.2004)

Inhaltsverzeichnis

1	Einleitung	1
2	Ziel der Diplomarbeit	3
3	HSM Konzepte	5
3.1	Konzept -DMF-	7
3.2	Konzept -TSM-	11
4	Untersuchungen	15
4.1	Konfiguration des IBM Supercomputers	15
4.2	Weitere Komponenten der Testumgebung	16
4.3	Wie lange dauert die Migration einer Datei?	16
4.4	Wie lange dauert der Recall einer Datei?	18
4.5	Wie verhält sich das Datenwachstum auf dem Dateisystem?	18
4.6	Zusammenhang zwischen Zugriffswahrscheinlichkeit und letztem Dateizugriff	19
4.6.1	Gibt es einen Zusammenhang zwischen der Zugriffswahrscheinlichkeit und der Zeit seit dem letzten Zugriff?	19
4.6.2	Welchen Anteil haben Dateien der verschiedenen Alterskategorien an der Menge der täglich zugegriffenen Dateien?	22
4.7	Zusammenhang zwischen Zugriffswahrscheinlichkeit und Größe	24
4.7.1	Welchen Anteil haben Dateien der verschiedenen Größenkategorien an der Menge der täglich zugegriffenen Dateien?	24
4.8	Zusammenhang zwischen Zugriffswahrscheinlichkeit, Dateigröße und letztem Zugriff	27
5	Ergebnisse	31
5.1	Wie kann immer freier Speicherplatz zur Abspeicherung von Dateien auf dem Dateisystem garantiert werden?	31
5.1.1	Schwellenwerte	31
5.1.2	Premigrationswert	32
5.2	Wie können die Zugriffszeiten für die Benutzer minimiert werden?	32
5.2.1	Kann eine Migrationsstrategie bzgl. des Alters einer Datei entwickelt werden?	32
5.2.2	Kann eine Migrationsstrategie bzgl. der Größe einer Datei entwickelt werden?	33
5.2.3	Einfluss der gewählten Migrationsstrategie auf die Kosten	34
5.2.4	Ergebnis: Migrationsstrategie	37
6	Ausblick	39
6.1	Übertragung der Migrationsstrategie auf den IBM Supercomputer Jump	39
6.2	Lebensdaueranalyse	40
A	Tabellen	41
A.1	1. Migrationsstrategie: Mindestdateialter	41

A.2	2. Migrationsstrategie: Mindestdateigröße	42
-----	---	----

Abbildungsverzeichnis

4.1	Bandspulzeiten	17
4.2	Schematischer Aufbau eines Bandes	17
4.3	Datenwachstum auf Dateisystemen FS1, FS2, FS3	19
4.4	Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS1)	21
4.5	Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS2)	21
4.6	Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS3)	22
4.7	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS1)	23
4.8	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS2)	23
4.9	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS3)	24
4.10	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS1)	25
4.11	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS2)	26
4.12	Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS3)	26
5.1	Recalldauer von Dateien	33
5.2	Durchschnittliche Dauer des Recallvorgangs für 100 MB in Abhängigkeit von der Dateigröße	34

Tabellenverzeichnis

4.1	Einteilung der Alterskategorien	20
4.2	Einteilung der Größenkategorien	24
4.3	Klasseneinteilung nach modifizierten Alterskategorien	27
4.4	Verteilung der Dateien des Dateisystems FS1 auf Alters- und Größenkategorien . .	27
4.5	χ^2 Werte der einzelnen Dateisysteme	29
4.6	Zugriffswahrscheinlichkeit (in %) auf Dateien des Dateisystems FS1 der verschie- denen Alters- und Größenkategorien	30
5.1	Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS1	36
5.2	Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS1	36
6.1	Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf den Dateisystem FS1-FS7	40
A.1	Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS2	41
A.2	Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS3	42
A.3	Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS2	42
A.4	Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS3	42

Zusammenfassung

Im Verlauf der nächsten Jahre wird auf dem neuen IBM-Supercomputer im Forschungszentrum Jülich ein großes Volumen an Benutzerdaten erwartet. Bei einem Großteil der Benutzerdaten handelt es sich um umfangreiche Dateien, die viel Speicherplatz benötigen und auf die, nachdem eine gewisse Zeitspanne vergangen ist, nur noch selten zugegriffen wird. Trotz der geringen Zugriffshäufigkeit liegen diese Dateien ebenso wie aktive Daten auf dem lokalen Dateisystem. Sie tragen dadurch zum rasanten Wachstum des Speicherbedarfs samt dem damit verbundenen Administrations- und Kostenaufwand bei.

Es ist kein sehr effizienter Umgang mit betrieblichen Ressourcen, wenn dieser hohe Aufwand unterschiedslos für inaktive Daten wie für häufig genutzte Dateien getrieben wird. Aus diesem Grund sollten nur letztere auf den schnellen Speichersystemen vorgehalten, erstere aber automatisch auf preisgünstigere Speichermedien ausgelagert werden (Migration). Genau diese Aufgabe leistet das Hierarchische Speichermanagement (HSM).

Ein Hierarchisches Speichermanagement System sollte möglichst effizient arbeiten. Dies zeichnet sich dadurch aus, dass immer ausreichend Speicherplatz zur Abspeicherung von Daten auf dem Dateisystem zur Verfügung steht und dass der Zugriff auf Daten möglichst schnell erfolgen kann.

In der Diplomarbeit werden diese beiden Aspekte betrachtet und untersucht. Es werden Kriterien vorgestellt, die besagen, wann eine Migration eingeleitet bzw. angehalten wird. Darüber hinaus wird untersucht, ob es Zusammenhänge zwischen der Zugriffswahrscheinlichkeit und dem Alter bzw. der Größe einer Datei gibt. Schließlich wird eine für den Jump Workload optimale Migrationsstrategie entwickelt.

Abstract

During the next years a large amount of user data will be expected on the new IBM Supercomputer of the Forschungszentrum Juelich. Most of these user data consist of large files, which need a lot of storage and which will be accessed seldom, after a certain time has elapsed. In spite of the rare frequency of access this data are filed - like active data - on the local filesystem. These data contribute to the rapid increase of storage requirements, with their efforts in money and administration. Operational resources aren't managed efficiently, if the same effort is made for inactive data and for frequently accessed data. For this reason only frequently accessed data should be filed on the filesystem and other data should be swapped out automatically onto cheaper storage medias (migration). Such a task is performed by a hierarchical storage management (HSM).

A hierarchical storage management system should work as efficiently as possible. There should always be sufficient storage for filing data on the filesystem and access to data should be as fast as possible.

In this diploma thesis these two aspects are considered. Criteria are shown, when to start a migration and when to stop it. Beyond this it is examined, if there are any coherences between the probability of access and the age respectively the size of a file. After all it is developed an optimal migrationstrategy for the jump workload.

Kapitel 1

Einleitung

Erfahrungen zeigen, dass ein hoher Anteil aller auf einem Dateisystem gespeicherten Benutzerdateien inaktiv ist, d.h. auf sie wird nur selten zugegriffen. Trotz der geringen Zugriffshäufigkeit liegen diese Dateien ebenso wie aktive Daten auf dem lokalen Dateisystem. Sie tragen dadurch zum rasanten Wachstum des Speicherbedarfs auf einem System samt dem damit verbundenen Administrations- und Kostenaufwand bei.

Es liegt auf der Hand, dass es kein sehr effizienter Umgang mit betrieblichen Ressourcen ist, wenn dieser hohe Aufwand unterschiedslos für inaktive Daten wie für häufig genutzte Dateien getrieben wird. Sowohl aus Kostengründen als auch unter dem Gesichtspunkt der Administration sollten nur letztere auf den schnellen Speichersystemen vorgehalten, erstere aber automatisch auf preisgünstigere Medien ausgelagert werden. Genau diese Aufgabe leistet das Hierarchische Speichermanagement (HSM).

Das Grundprinzip von HSM ist das automatische Verschieben von Daten auf verschiedene Ebenen einer Speicherhierarchie nach bestimmten, vom Systemadministrator definierten Kriterien. Dahinter steht der Gedanke, dem Nutzer für seine aktuellen Anwendungen ein Maximum an Speicherkapazität auf dem in seinem System schnellsten Medium, in der Regel dem Festplattenspeicher, bereitzustellen. Dazu werden Dateien auf die jeweils nächst tiefere Ebene der Speicherhierarchie verschoben, sobald bestimmte Bedingungen erfüllt sind. Diesen Vorgang bezeichnet man als Migration. Die Migration läuft vollständig automatisch ab.

Für den Benutzer ist der Vorgang transparent. Die Dateien bleiben im Dateiverzeichnis aufgelistet und der Benutzer kann auf migrierte Dateien zugreifen, als ob sie im Dateisystem abgelegt wären. Wird auf eine migrierte Datei zugegriffen, bringt das System die Datei automatisch in das Dateisystem zurück (Recall).

Mittels des Hierarchischen Speichermanagement kann man eine optimale Nutzung der vorhandenen Speicherressourcen erreichen und dem Benutzer virtuell praktisch unbegrenzte Speicherkapazitäten zur Verfügung stellen ohne Investitionen in neue, leistungsfähigere Hardware zu machen.

Darüber hinaus lässt sich mit HSM das Datenwachstum und die damit verbundene Zunahme der Speicherkapazitäten leichter und kostengünstiger abfangen als auf andere Weise.

Kapitel 2

Ziel der Diplomarbeit

Der neue IBM Supercomputer im Forschungszentrum Jülich steht für Projekte des wissenschaftlichen Rechnens zur Verfügung. Im Verlauf der nächsten Jahre wird ein großes Volumen an Benutzerdaten erwartet. Bei einem Großteil der Benutzerdaten handelt es sich um umfangreiche Dateien, die von Programmen im Batch Betrieb eingelesen bzw. ausgegeben werden. Da die Daten viel Speicherplatz benötigen und die Erfahrung mit Cray-Supercomputing zeigt, dass auf die Daten, nachdem eine gewisse Zeitspanne vergangen ist, nur noch selten zugegriffen wird, ist es aus Kosten- und Administrationsgründen nicht sinnvoll, alle Daten auf dem lokalen Dateisystem zu halten. Aus diesem Grund wird ein Hierarchisches Speichermanagement eingesetzt und ein Teil der Daten auf eine zweite Speicherhierarchieebene kopiert. Die erste Ebene wird durch Platten realisiert. Der Zugriff auf sie ist schneller als auf Bänder, welche die Speichermedien der zweiten Ebene sind. Ein solches System soll möglichst effizient arbeiten. Ein effizientes Hierarchisches Speichermanagement System zeichnet sich dadurch aus, dass immer ausreichend Speicherplatz zur Abspeicherung von Daten auf dem Dateisystem zur Verfügung steht und dass der Zugriff auf Daten möglichst schnell erfolgen kann.

Im Rahmen der Diplomarbeit sollen folgende zwei Fragestellungen untersucht werden:

- Wie kann immer freier Speicherplatz zur Abspeicherung von Daten auf dem Dateisystem garantiert werden?
- Wie können die Zugriffszeiten für den Benutzer minimiert werden?

Würden die beiden Fragen unabhängig voneinander betrachtet werden, so würden im ersten Fall alle Dateien mit Ausnahme einiger Systemdateien, die immer auf dem lokalen Dateisystem gehalten werden müssen, auf die zweite Speicherhierarchieebene kopiert, um möglichst viel freien Platz auf dem Dateisystem zu haben. Im zweiten Fall würden alle Dateien auf dem lokalen Dateisystem gehalten werden, da der Zugriff auf Daten, die auf dem lokalen Dateisystem gehalten werden, am schnellsten ist. Betrachtet man diese beiden Antworten, so lässt sich ein klarer Widerspruch erkennen.

Ziel der Diplomarbeit ist es, beide Gesichtspunkte **gleichzeitig** zu betrachten und zu optimieren.

Wie kann immer freier Speicherplatz zur Abspeicherung von Daten auf dem Dateisystem garantiert werden?

Es werden folgende Messungen und Auswertungen durchgeführt:

- Wie lange dauert die Migration einer Datei?
Es wird die Mountzeit eines Bandes, die Übertragungsrate und die Spulzeit bestimmt.
- Wie verhält sich das Datenwachstum auf dem Dateisystem?
Der Anstieg des Datenvolumens wird statistisch ausgewertet.

Abhängig von der Migrationsdauer und dem Datenwachstum wird dann die Höhe des Füllungsgrades des Dateisystems, bei der die Migration eingeleitet bzw. angehalten wird, festgelegt, so dass im Normalfall immer ausreichend Platz zur Abspeicherung von Daten vorhanden ist.

Wie können die Zugriffszeiten für die Benutzer minimiert werden?

Die Zeit, die benötigt wird, eine einzelne migrierte Datei auf das Dateisystem zurückzuholen, kann nicht verkürzt werden. Es kann nur die Summe aller Zugriffszeiten minimiert werden, wenn die Dateien migriert werden, auf die seltener zugegriffen wird. Deswegen wird in dieser Arbeit untersucht, ob es einen Zusammenhang zwischen der Zugriffswahrscheinlichkeit und dem letzten Dateizugriff bzw. der Dateigröße gibt. Existiert ein solcher Zusammenhang, dann ist es sinnvoll zuerst die Dateien zu migrieren, für die die Zugriffswahrscheinlichkeit gering sind. Reicht der auf diese Weise geschaffene Speicherplatz aus, so müssen keine weiteren Dateien migriert werden. Ansonsten werden auch Dateien migriert, auf die häufiger zugegriffen wird.

Unabhängig von der Zugriffswahrscheinlichkeit ist es sinnvoll, bei der Migration mit den großen Dateien zu beginnen. Zum einen müssen dann weniger Dateien migriert werden, was zur Folge hat, dass die Wahrscheinlichkeit, dass ein Benutzer auf eine migrierte Datei zugreift, sinkt. Zum anderen fällt die Verzögerung, die durch das Mounten und Spulen eines Bandes beim Zurückholen einer Datei entsteht, weniger stark ins Gewicht als bei einer kleinen Datei.

Es werden folgende Untersuchungen gemacht:

- Wie lange dauert es, eine Datei auf das lokale Dateisystem zurückzuholen?
- Lässt sich verifizieren, dass je länger auf eine Datei nicht zugegriffen wurde, desto geringer ist die Wahrscheinlichkeit, dass in Zukunft auf sie zugegriffen wird?
- Lässt sich verifizieren, dass je größer eine Datei ist, desto geringer ist die Wahrscheinlichkeit, dass in Zukunft auf sie zugegriffen wird?
- Kann eine Migrationsstrategie bzgl. Alter und Größe einer Datei entwickelt werden?

Kapitel 3

Beschreibung von Hierarchischen Speichermanagement Systemen

Um die Funktionalität und die Arbeitsweise von Hierarchischen Speichermanagement Systemen zu verdeutlichen, werden im folgenden zwei verschiedene Hierarchische Speichermanagement Systeme beschrieben.

Zunächst wird die Cray Data Migration Facility erläutert, die auf den bestehenden Cray Systemen T3E-1200 und SV1ex genutzt wird. Anschliessend wird der Hierarchical Storage Manager von IBM Tivoli Storage Management erklärt, der auf dem IBM Supercomputer installiert ist.

3.1 Cray Data Migration Facility (DMF)

Die Cray Data Migration Facility (DMF) sichert die Verfügbarkeit von freiem Speicherplatz auf dem lokalen Dateisystem, indem Dateien anhand bestimmter Kriterien ausgewählt, auf ein externes Speichermedium kopiert und auf dem lokalen Dateisystem gelöscht werden.

Die DMF setzt einen Daemon ein, der zur Kommunikation zwischen Kommandos, *medienspezifischen Prozessen* und dem Betriebssystemkern dient. Als *medienspezifischen Prozess* bezeichnet man einen Prozess, der Daten vom lokalen Dateisystem auf externe Speichermedien wie Bänder oder entfernte Rechnersystem schreibt bzw. von ihnen liest. Der Betriebssystemkern stellt eine Art Vermittler zwischen den Anwenderprogrammen und der Hardware des Computers dar. Er bietet für unterschiedliche Hardware einheitliche Schnittstellen an, um auf die Systemkomponenten zugreifen zu können.

Darüber hinaus steuert der Daemon das Abspeichern von Metadaten zu den auf externen Speichermedien kopierten Dateien in einer relationalen Datenbank. Werden Dateien auf ein externes Speichermedium kopiert, dann modifiziert der Daemon deren Inode, indem er zum einen den Status jeder Datei aktualisiert und zum anderen die Information, wo die Metadaten zu finden sind, hinzufügt.

Bei DMF unterscheidet man zwischen einem *Standard Modus* und einem *Client/Server Modus*.

Beim *Standard Modus* steuert ein einziger Daemon den Migrations- und Recallprozess. Es existiert eine DMF Konfigurationsdatei für den Daemon.

Den *Client/Server Modus* setzt man ein, wenn verschiedene Cray Systeme auf dasselbe Dateisystem zugreifen und Daten von diesem migriert werden. In diesem Fall muss auf jedem Cray System ein DMF Client Daemon laufen und auf mindestens einem Cray System muss ein DMF Server Daemon laufen. Der Client Daemon nimmt Anfragen des Kernels und der Benutzerprozesse entgegen und sendet sie an den Server Daemon. Dieser steuert einen *medienspezifischen Prozess*, um Daten auf ein externes Speichermedium zu migrieren oder von diesem zurückzuholen. In diesem Modus gibt es für jeden Client und dem Server eine DMF Konfigurationsdatei.

Eine Datei wird entweder von einem Benutzer explizit durch das Absetzen eines Kommandos oder automatisch durch eine Speichermanagementkomponente von DMF migriert.

Automatische Migration

Die automatische Migration wird durch Speichermanagementkomponenten vom DMF Daemon angestoßen, die das Dateisystem überwachen. Es werden folgende Schwellenwerte spezifiziert:

- *Critical Threshold*
- *Warning Threshold*
- *Free Space Threshold*
- *Migration Threshold*

Die Schwellenwerte sollen wie oben aufgelistet vom *Critical Threshold* bis hin zum *Migration Threshold* einen immer kleiner werdenden belegten Speicherplatz präsentieren. Erreicht der Füllungsgrad des Dateisystems den *Warning Threshold*, dann wird ein Skript ausgeführt, das die beiden Kommandos *dmmctl* und *fsmon* aufruft.

Das Kommando *fsmon* bewirkt, dass ein Dateisystem weiterhin überwacht wird. Das Skript kann allerdings erst wieder ausgeführt werden, wenn der Füllungsgrad des Dateisystems unter den Schwellenwert gesunken ist.

Das Kommando *dmmctl* bewirkt folgende Aktionen: Zunächst wird für jede Datei im Dateisystem eine positive Maßzahl anhand einer Berechnungsformel bestimmt. In die Berechnungsformel gehen zwei Kriterien ein: die Zeit, die seit dem letzten Zugriff auf die Datei vergangen ist und die

Dateigröße. Durch Setzen von Faktoren kann der Administrator festlegen, wie stark die beiden Kriterien gewichtet werden. Die Dateien und ihre berechnete Maßzahl, die die Migrationsreihenfolge festlegt, werden abfallend in einer *Migrationsliste* abgespeichert.

Nachdem diese Rangliste gebildet wurde, werden Dateien, die schon zu einem früheren Zeitpunkt auf ein externes Medium geschrieben wurden, sich aber auch auf dem lokalen Dateisystem befinden, betrachtet. Sind die Daten auf dem externen Medium mit denen auf dem lokalen Dateisystem identisch, so wird der Speicherplatz freigegeben.

Dieser Vorgang läuft solange ab, bis der *Free Space Threshold* erreicht wird oder es keine weiteren Dateien auf dem Dateisystem gibt, von denen eine identische Datei auf Band existiert. Ist der erste Fall eingetreten, dann ist die Migration beendet. Ist der andere Fall eingetreten und der *Free Space Threshold* nicht erreicht worden, so wird die zuvor bestimmte *Migrationsliste* sequentiell durchlaufen. Die Dateien werden nacheinander auf externe Speichermedien kopiert und der Speicherplatz wird freigegeben, bis der *Free Space Threshold* erreicht wird. Dabei werden Daten erst auf ein Band geschrieben, wenn entweder eine bestimmte Anzahl Bytes für die Migration verfügbar ist oder wenn eine bestimmte Zeit seit dem letzten Bandmount vergangen ist. Ist keines der beiden Kriterien erfüllt und werden weiterhin Daten in das Dateisystem geschrieben, so kann es passieren, dass der Füllungsgrad des Dateisystems weiter ansteigt. Wird der *Critical Threshold* erreicht, dann wird ein weiteres Skript ausgeführt. Dieses enthält die Kommandos *dmmctl*, *fsmon* und *dmdidle*. Das Kommando *dmdidle* bewirkt, dass Dateien auch auf Band geschrieben werden, wenn die minimale Anzahl Bytes nicht erreicht ist.

Nachdem der *Free Space Threshold* erreicht wurde, werden weiterhin solange Dateien auf Band kopiert, jedoch auf dem Dateisystem nicht gelöscht, bis der *Migration Threshold* erreicht wird.

Sowohl der Administrator als auch der Benutzer können Einfluss nehmen, ob eine Datei migriert werden darf:

- Eintrag einer negativen Maßzahl für die Datei in der *Migrationsliste*.
Möchte der Administrator, dass eine Datei eine Mindestgröße hat oder dass seit dem letzten Zugriff auf die Datei eine Mindestanzahl an Tagen vergangen ist, bevor sie migriert werden darf, dann kann er für die Berechnung der Maßzahl festlegen, dass für Dateien, die diesen Kriterien nicht entsprechen, ein negativer Wert in der *Migrationsliste* eingetragen wird. Dateien, die einen negativen Wert in der Migrationsliste haben, werden nicht migriert.
- Der Benutzer trägt die Datei in der *.keep* Datei ein.
Jeder Benutzer kann ein maximales Datenvolumen von der Migration ausschließen. Die Menge des Datenvolumens wird vom Administrator festgelegt. Der Benutzer trägt die Dateien, die nicht migriert werden sollen in eine *.keep* Datei ein. Werden Kandidaten für die Migration ausgewählt, so werden zunächst alle Dateien in der *.keep* Datei bis zu diesem Maximum von der Migration ausgeschlossen. Ist das Maximum erreicht, so werden die weiteren Dateien, die in der *.keep* Datei stehen, unter Umständen migriert.

Recall einer migrierten Datei

Wird auf eine migrierte Datei zugegriffen, dann werden die Daten auf das lokale Dateisystem zurückgeholt.

Eine migrierte Datei, die auf das Dateisystem zurückgeholt wurde, wird als *Dual State File* bezeichnet, solange sie nicht modifiziert wurde. Wird ein solcher *Dual State File* wieder migriert, so muss keine neue Kopie der Datei erstellt werden, weil die existierende Datei noch gültig ist. Wird ein *Dual State File* modifiziert, wird der Eintrag in der Datenbank gelöscht.

Der Benutzer kann anhand eines *dmmode* Flags festlegen, was passiert, wenn ein Prozess eine migrierte Datei öffnet. Das *dmmode* Flag kann wie folgt gesetzt sein:

- *dmmode* gleich 0, ein automatischer Recall für migrierte Dateien ist nicht aktiviert. Das bedeutet, dass bei dem Versuch eine migrierte Datei zu öffnen, diese nicht automatisch auf das lokale Dateisystem zurückgeholt wird. Der Benutzer muss die Datei explizit durch Absetzen eines Kommandos zurückholen.
- *dmmode* gleich 1, ein automatischer Recall für migrierte Datei ist aktiviert. Bei dem Versuch eine migrierte Datei zu öffnen, wird zunächst der Prozess, der das Öffnen der Datei initiiert, blockiert und der DMF Deamon holt die Datei auf das lokale Dateisystem zurück. Wenn der Recall abgeschlossen ist, läuft der Prozess weiter und öffnet die Datei, die sich nun auf dem lokalen Dateisystem befindet.

Das Migrieren und Zurückholen von Dateien ist für den Benutzer transparent, denn die Daten bleiben im Dateiverzeichnis und werden behandelt als wären sie im Dateisystem abgelegt. Setzt man beispielsweise das 'ls' Kommando ohne weitere Optionen ab, so kann man keinen Unterschied zwischen einer migrierten und nicht migrierten Datei sehen.

3.2 Tivoli Storage Manager (TSM)

Das Hierarchische Speichermanagement von IBM Tivoli Storage Management (HSM) ist als Client-Server Anwendung implementiert. Der Server steuert die Ablage der Daten auf dem TSM Speicher. Der TSM Speicher besteht aus sogenannten *Speicherpools*. *Speicherpools* sind Gruppen von *Speichermedien* mit wahlfreiem oder sequentielltem Zugriff, die für das Auslagern von Dateien verwendet werden. *Speicherpools* können hierarchisch organisiert werden, d.h. der Administrator kann beispielsweise festlegen, dass die Speicherhierarchie aus zwei Ebenen besteht, wobei die erste aus Platten und die zweite aus Bändern besteht. Die Anzahl der Speicherhierarchieebenen lässt sich beliebig weit ausbauen.

Der Client sendet und empfängt Daten zum und vom Server. Er hat Zugriff auf eine relationale Datenbank, in der Metadaten zu den im TSM Speicher abgelegten Daten abgespeichert sind.

Wird eine Datei migriert, dann sendet der HSM Client eine Kopie der Datei an den TSM Server und ersetzt die Originaldatei auf dem lokalen Dateisystem durch einen *Stub File*.

Ein *Stub File* ist eine kleine Datei, die außer der Inode Information auch die nötige Information enthält, um eine migrierte Datei auf das lokale Dateisystem zurückzuholen (recall). Aufgrund der auf dem Dateisystem verbleibenden Inode Information muss beim Aufruf von bestimmten Unix Kommandos, wie beispielsweise bei einem 'ls', eine Datei nicht zurückmigriert werden.

Darüber hinaus kann der Administrator die Anzahl der Bytes, die von einer migrierten Datei auf dem lokalen Dateisystem gehalten werden, bestimmen.

Bei der Migration werden die Zeit des letzten Zugriffs und die Zugriffsrechte der Datei nicht geändert.

HSM unterstützt sowohl *automatische* als auch *selektive Migration*. Die *automatische Migration* wird auch als *Schwellenwertmigration* bezeichnet.

Zustand einer Datei

Es werden folgende Termina benutzt, um den Status einer Datei zu beschreiben.

resident: Eine residente Datei befindet sich auf dem lokalen Dateisystem. Eine neu angelegte Datei ist beispielsweise eine residente Datei.

migriert: Eine migrierte Datei ist eine Datei, die vom lokalen Dateisystem zum TSM Speicher kopiert und durch einen *Stub File* ersetzt wurde.

premigriert: Eine premigrierte Datei ist eine Datei, die vom lokalen Dateisystem zum TSM Speicher kopiert jedoch nicht durch einen *Stub File* ersetzt wurde. Eine identische Kopie einer Datei existiert auf dem lokalen Dateisystem und im TSM Speicher. Des weiteren hat eine Datei hat den Status premigriert, wenn sie recalled und nicht verändert wurde.

Schwellenwertmigration

Durch diese Form der Migration wird versucht, zu jedem Zeitpunkt mindestens einen minimalen freien Speicherplatz auf dem lokalen Dateisystem zu gewährleisten.

Für das Dateisystem werden *oberer* und *unterer Schwellenwert* festgelegt. Diese Schwellenwerte sind prozentuale Angaben, die sich auf den Füllungsgrad des lokalen Dateisystems beziehen. Sie besagen, dass die Migration eingeleitet bzw. angehalten wird, wenn der Füllungsgrad eine festgelegte Marke über- bzw. unterschreitet.

Der Client kontrolliert die Speicherplatzbelegung des lokalen Dateisystems in bestimmten vom Administrator festgelegten Intervallen. Wenn die Speicherplatzbelegung den *oberen Schwellenwert* erreicht, so werden automatisch die Dateien migriert, die zur Migration zur Verfügung stehen.

Der Administrator kann Kriterien festlegen, die erfüllt sein müssen, damit eine Datei für eine Migration zur Verfügung steht. Beispielsweise kann er fordern, dass eine festgelegte Anzahl von Tagen seit dem letzten Zugriff vergangen sein muss, die Datei eine bestimmte Größe haben muss oder die Datei schon im *Backup* liegt, d.h. dass eine Sicherheitskopie von der Datei existiert.

Für alle migrierbaren Dateien erstellt der HSM Client eine *Kandidatenliste*, um die Reihenfolge, in der die Dateien migriert werden, festzulegen. Der Administrator kann festlegen, dass die Priorität einer Datei anhand der Zeit, die seit dem letzten Zugriff vergangen ist und anhand der Dateigröße bestimmt wird. Darüber hinaus kann ein *Zeit-* und ein *Größenfaktor* festgelegt werden, um das Zeit- und Größenkriterium unterschiedlich zu gewichten.

Der Migrationsvorgang hält solange an, bis der *untere Schwellenwert* für das Dateisystem erreicht ist oder keine Datei mehr zur Migration zur Verfügung steht.

Der Administrator kann jedoch *Schwellenwertmigration* auch einleiten, wenn der *obere Schwellenwert* noch nicht erreicht wurde. Bei dieser manuellen Einleitung der *Schwellenwertmigration* werden Dateien solange migriert, bis der *untere Schwellenwert* unterschritten wird oder keine Dateien mehr vorhanden sind, die migriert werden können.

Alle 10 Sekunden überprüft HSM, ob der Speicherplatz auf dem Dateisystem voll läuft. Ist dies der Fall, so wird der Prozess, der diesen Zustand verursacht hat, gestoppt und automatisch die *Schwellenwertmigration* eingeleitet. Wenn es wieder ausreichend freien Speicherplatz auf dem Dateisystem gibt, wird der zuvor gestoppte Prozess weiter fortgeführt. Der Prozess muss nicht warten bis die *Schwellenwertmigration* abgeschlossen ist, und es wird keine out-of-space message ausgegeben.

Premigration

Um die Migration zu beschleunigen, verwendet HSM den Prozess der *Premigration*. HSM kopiert dabei Dateien in den TSM Speicher, lässt aber die Originaldatei unverändert auf dem lokalen Dateisystem. Diese Dateien haben den Zustand premigriert. Wenn freier Speicherplatz auf dem Dateisystem geschaffen werden muss, dann kontrolliert HSM, ob die Dateien seit der *Premigration* nicht verändert wurden. Ist dies der Fall, dann werden die Kopien auf dem lokalen Dateisystem durch *Stub Files* ersetzt. Auf diese Weise können premigrierte Dateien direkt in den Status migriert gesetzt werden, ohne dass sie zum TSM Speicher kopiert werden müssen.

Nach jeder *automatischen Migration* wird eine *Premigration* durchgeführt. Es werden so viele Dateien in den Status premigriert gebracht, bis der *Premigrationswert*, der für das Dateisystem gesetzt ist, erreicht ist. Der *Premigrationswert* beschreibt den prozentualen Anteil des Dateisystems, den man als Vorbereitung für die nächste Migration premigrieren möchte. Die Defaulteinstellung für den *Premigrationswert* ist die Differenz zwischen dem *oberen* und *unteren Schwellenwert* des Dateisystems. Der Administrator kann den *Premigrationswert* jedoch jederzeit verändern.

Selective Migration

Der Benutzer kann *Selective Migration* benutzen, um Dateien, auf die er längere Zeit nicht mehr zugreifen wird, auf den TSM Speicher auszulagern. Bei der *Selectiven Migration* migriert HSM nur Dateien, die zur Migration freigegeben sind. Der Administrator kann die gleichen Kriterien wie bei der *Schwellenwertmigration* festlegen jedoch mit der Ausnahme, dass bei der *selektiven Migration* das letzte Zugriffsdatum nicht berücksichtigt wird.

Recalling von migrierten Dateien

Als *Recall* wird das Zurückholen von migrierten Dateien auf das Dateisystem bezeichnet. Jedes Mal, wenn HSM eine Datei zurückholt, wird die Zeit des letzten Zugriffs für die Datei auf die aktuelle Zeit gesetzt.

HSM bietet *automatische* und *selektive Recalls*.

Automatische Recalls

Um auf eine migrierte Datei zuzugreifen, muss der Benutzer nicht wissen, dass die Datei migriert ist. Greift er auf eine migrierte Datei zu, dann bringt ein *Transparenter Recall* die Datei automatisch auf das lokale Dateisystem zurück.

Wenn HSM eine migrierte Datei zurückholt, dann bleibt die Kopie im TSM Speicher zurück. Weil eine identische Kopie einer Datei auf dem lokalen Dateisystem und im TSM Speicher existiert, hat die Datei den Status premigriert. Die Datei bleibt solange premigriert, bis sie erneut migriert oder verändert wurde. Wird eine Datei verändert, so erhält sie den Status resident File.

Der Administrator kann eine Option setzen (die KERNELMESSAGE Option), die bestimmt, ob HSM den Benutzer informiert, wenn eine Datei in das lokale Dateisystem zurückgeholt wird. Ein Prozess kann jedoch nicht erkennen, ob eine Datei auf dem lokalen Dateisystem oder im TSM Speicher liegt.

Selective Recalls

Der Benutzer kann über das graphische Benutzerinterface von HSM oder mittels HSM Kommandos herausfinden, ob eine Datei migriert ist oder nicht. Ist eine Datei migriert, so kann er sie auf das Ursprungsdateisystem zurückholen.

Abgleichen von Dateisystem und TSM Speicher

Um die Daten, die auf dem lokalen Dateisystem und dem TSM Speicher liegen, konsistent zu halten, gleicht der HSM Client das Dateisystem mit dem TSM Speicher in regelmäßigen Intervallen ab, die durch den Administrator festgelegt werden.

Greift der Benutzer auf eine migrierte oder premigrierte Datei zu und verändert sie oder löscht er sie vom lokalen Dateisystem, dann liegt im TSM Speicher eine nicht aktuelle Kopie. Während des Abgleichens vom lokalen Dateisystem und TSM Server markiert TSM jede veraltete Kopie einer migrierten oder premigrierten Datei als löscherbar. Die Kopien werden nach einer festgelegten Zeit vom Server gelöscht.

Dateien vom Speichermanagement ausschliessen

Bestimmte Dateien, wie zum Beispiel Systemdateien und Dateien, die von HSM erstellt und benutzt werden, schließt HSM automatisch von der Migration aus.

Möchte der Benutzer, dass weitere Dateien von der Migration ausgenommen werden, so kann er in einer Datei, dem *Include-exclude Option File*, Dateien oder Gruppen von Dateien eintragen, die nicht migriert werden sollen.

Setzen der Speichermanagementoptionen

Der Administrator kann Speichermanagementoptionen auf verschiedene Weisen setzen:

1. in der *dsm.sys* Datei,
2. durch Einstellungen beim Server,
3. durch ein Kommando.

Im Folgenden werden diese drei Möglichkeiten näher vorgestellt.

1. In der *dsm.sys* Datei kann der Administrator unter anderem folgende Einstellungen festlegen:
 - zu welchem TSM-Server Daten migriert und premigriert werden,
 - wie oft HSM die Speicherplatzbelegung des Dateisystems kontrolliert,
 - die minimale Größe, die eine Datei haben muss, um zur Migration verfügbar zu sein,
 - in welchen Intervallen das lokale Dateisystem und der TSM Server synchronisiert werden,
 - wieviel Zeit vergehen muss, nachdem eine Datei auf dem lokalen Dateisystem geändert oder gelöscht wurde, bevor die Kopie im TSM Speicher gelöscht wird,
 - die maximale Anzahl an Migrationsprozessen, die auf einem Dateisystem gleichzeitig stattfinden kann,
 - die maximale Anzahl an Recalldaemons, die gleichzeitig auf einem Client-Knoten laufen können.
2. Beim Server kann man folgende Einstellungen vornehmen:
 - ob HSM eine Datei automatisch und bei Benutzeraufforderung, nur bei Benutzeraufforderung oder gar nicht migrieren kann,
 - die Anzahl der Tage, die seit dem letzten Zugriff vergangen sein muss, damit eine Datei migriert werden kann,
 - ob eine aktuelle *Backupversion* der zu migrierenden Datei auf dem Server, zu dem migriert wird, existieren muss, bevor eine Datei zur Migration zur Verfügung steht,
 - auf welche Gruppen von Platten oder Bändern die Datei abgespeichert wird, wenn sie migriert wird.
3. Durch ein Kommando wird festgelegt:
 - der *obere* und *untere Schwellenwert* für den Füllungsgrad eines Dateisystems, (Der *obere* und *untere Schwellenwert* bestimmt, wann HSM automatisch eine Migration einleitet und beendet.)
 - der Anteil eines Dateisystems, der premigriert wird (*Premigrationswert*),
 - die Größe des *Stub Files*, der auf einem Dateisystem nach einer Migration zurückbleibt, (Die *Stub File* Größe bestimmt, wie viele Bytes vom Anfang einer migrierten Datei im *Stub File* stehen. Wenn nur auf die Daten, die im *Stub File* stehen, zugegriffen wird und diese nicht verändert werden, so muss die Datei nicht in den TSM Speicher zurückgeholt werden. Die *Stub File* Größe bestimmt auch die Mindestgröße einer Datei, damit sie zur Migration zur Verfügung steht. Um zur Migration zur Verfügung zu stehen, muss eine Datei mindestens so groß sein wie die angegebene *Stub File* Größe plus 1 Byte und wie die Blockgröße, die für das Dateisystem festgelegt ist.)
 - die Reihenfolge, in der zur Verfügung stehende Dateien migriert werden, (Der Administrator kann einen *Zeit-* bzw. *Größenfaktor* festlegen, um die Zeit, die sei dem letzten Zugriff vergangen ist, und die Größe einer Datei unterschiedlich stark zu gewichten.)
 - das maximale Datenvolumen, das von einem Dateisystem migriert und premigriert werden kann, (Die Zahl wird auch *Quota* genannt. Die Defaulteinstellung ist gleich der Dateisystemgröße.)

Kapitel 4

Untersuchungen

Die Untersuchungen werden auf dem IBM Supercomputer der Stufe 0.5 ¹ des Forschungszentrum Jülich gemacht.

4.1 Konfiguration des IBM Supercomputers

Der IBM Supercomputer der Stufe 0.5 des Forschungszentrum Jülich ist eine Clusterarchitektur bestehend aus sechs Knoten IBM p690. Jeder Knoten ist mit 32 Prozessoren des Typs Power4+ mit 1,7 GHz Taktrate und 64 GByte gemeinsamen Hauptspeicher ausgestattet. Das System erreicht damit eine Peak-Leistung von 1,3 TFLOPS.

Ein Knoten ist als Login-Knoten konfiguriert. Er ist in vier logische Partitionen unterteilt, d.h. dass dieser Knoten in mehrere unabhängige logische Teilsysteme aufgeteilt ist, die sich durch ihre Hard- und Softwarekomponenten unterscheiden.

Eine Partition ist die Login-Partition, eine weitere Partition wird für Datamanagement-Aufgaben wie z.Bsp. I/O, Backup und Hierarchisches Speichermanagement genutzt. Die beiden weiteren logischen Partitionen werden für Testzwecke verwendet.

Die anderen fünf Knoten sind als Compute-Nodes konfiguriert und für die Ausführung paralleler Programme vorgesehen. Auf einem der fünf Knoten laufen interaktive Anwendungen und auf den anderen vier Knoten laufen Batch Jobs.

Die Knoten des Systems sind über ein Gigabit-Ethernet miteinander verbunden.

Da dies bei der knotenübergreifenden Ausführung paralleler Programme zu erheblichen Kommunikationsengpässen und somit zu Performance-Verlusten führen würde, können Programme zur Zeit nur maximal einen Knoten, d.h. 32 Prozessoren anfordern.

Das System wird unter dem Betriebssystem AIX 5.1 betrieben. Bei einem Login auf das System gelangt der Benutzer automatisch auf die Login-Partition. Diese Partition steht für die Vorbereitung, die Submission und die Nachbearbeitung von Batch-Jobs sowie die Ausführung interaktiver Anwendungen zur Verfügung.

Sämtliche Benutzerdaten werden in fünf HOME-Filesystemen unter Kontrolle von Global Parallel File System (GPFS) gespeichert. GPFS ist ein cluster-weites Dateisystem, das den gemeinsamen Zugriff von Benutzern auf Dateien ermöglicht, die mehrere Plattenlaufwerke überspannen. GPFS ermöglicht den Zugriff auf Platten, die nicht direkt an die Anwendungsknoten angeschlossen sind und gewährleistet die Datenkohärenz beim Zugriff von einem beliebigen Knoten.

¹Vorstufe der Gesamtkonfiguration des IBM Rechners

4.2 Weitere Komponenten der Testumgebung

Auf der Partition, die für Datamanagement-Aufgaben genutzt wird, sind der HSM Client und Server Version 5.2 installiert.

An den Supercomputer ist ein Bandroboter angeschlossen. Dieser hat 16 Bandstationen. Insgesamt stehen 5608 Zellen für Bänder zur Verfügung, davon sind ca. 3000 belegt. Die Bänder haben eine Kapazität von 200 GB. Geht man von den 3000 derzeit zur Verfügung stehenden Bändern aus, so können 600 TB unkomprimiert auf Band geschrieben werden.

Zur statistischen Auswertung des Datenwachstums und zur Prüfung, ob es einen Zusammenhang zwischen Dateialter bzw. Dateigröße und Zugriffshäufigkeit gibt, werden die Dateien, die sich auf den HOME-Dateisystemen befinden, ausgewertet. Von den fünf Home-Dateisystemen, die jeweils eine Kapazität von 1 TB haben, werden drei Dateisysteme betrachtet.

Darüber hinaus gibt es ein Dateisystem, das zu HSM Testzwecken genutzt wird. Das Dateisystem hat eine Größe von 546 GB.

Zur Bestimmung der Migrations- und Recallzeiten werden Daten von diesem Dateisystem auf Bänder des oben beschriebenen Roboters migriert bzw. zurückgeholt.

4.3 Wie lange dauert die Migration einer Datei?

Wird eine Datei auf ein Band geschrieben, so muss das Band zunächst gemountet werden. Anschließend wird das Band an die Schreibposition gespult. Die Daten werden dann auf das Band geschrieben.

Mountzeit

Zur Bestimmung der Mountzeit werden einhundert 1 KB große Dateien nacheinander auf ein leeres Band geschrieben und die Zeit gemessen. Bei jedem Schreibvorgang wird das Band neu gemountet. Die Mountzeit kann auf diese Weise gemessen werden, da die Spulzeit und Übertragungsrates für eine 1 KB große Datei sehr klein ist, wie sich im folgenden noch zeigen wird.

Die Messungen ergaben, dass es durchschnittlich *35 Sekunden* dauert, ein Band zu mounten.

Spulzeit

Zur Bestimmung der Spulzeit wird ein leeres Band mit jeweils 1 GB großen Dateien vollgeschrieben und die Zeit für jeden einzelnen Vorgang gemessen. Das Band wird für jeden Schreibvorgang neu gemountet und automatisch an die nächste Schreibposition gespult. Auf diese Weise wird die Zeit für das Mounten und Spulen des Bandes sowie die Datenübertragungsdauer gemessen. Subtrahiert man von dieser Zeit die Mount- und Übertragungsdauer, so erhält man folgende Bandspulzeiten, die in der Abbildung 4.1 dargestellt sind. Es lässt sich eine mittlere Bandspulzeit von *58 Sekunden* bestimmen. Die einzelnen gemessenen Werte ergeben zusammen eine wellenförmige Kurve, da die Speicherung der Daten in einem serpentinenförmigen Aufzeichnungsverfahren parallel zum längsseitigen Bandrand erfolgt und sich die Schreib- und Leseinheit senkrecht dazu über das Band bewegt. Die Abbildung 4.2 zeigt den Schreib- und Lesemechanismus auf ein Band.

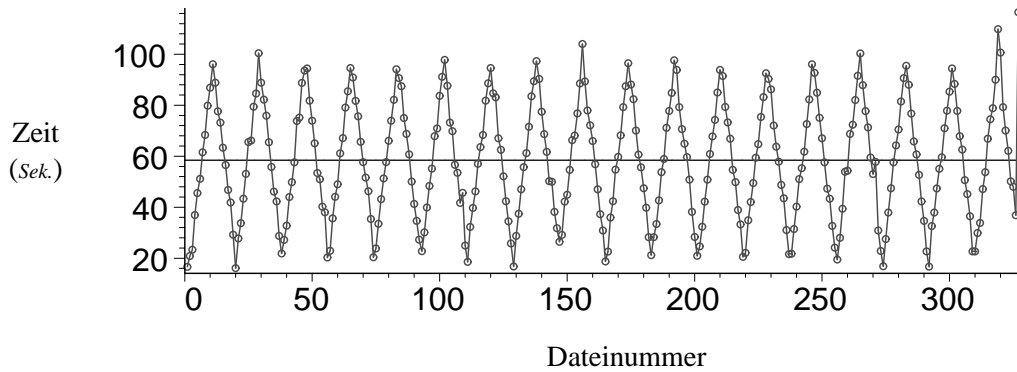


Abbildung 4.1: Bandspulzeiten

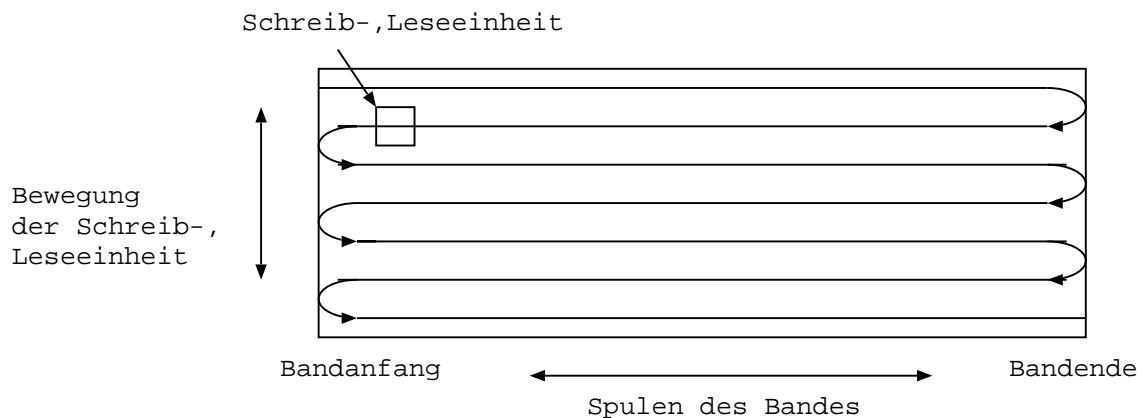


Abbildung 4.2: Schematischer Aufbau eines Bandes

Datenübertragungsrate

Zur Bestimmung der Datenübertragungsrate werden einhundert 1 GB große Dateien nacheinander auf ein leeres Band geschrieben und die Zeit gemessen. Zwischen den einzelnen Schreibvorgängen wird das Band nicht neu gemountet.

Die Messungen ergaben eine durchschnittliche Übertragungsrate von 30 MB/s . Folglich dauert es ca. 34 Sekunden, 1 GB zu migrieren.

Der maßgebende Faktor, der die Datenübertragungsrate beschränkt, ist die Geschwindigkeit, mit der eine Bandstation ein Band beschreiben kann. Eine Bandstation kann 30 MB in einer Sekunde auf ein Band schreiben. Die tatsächliche Datenübertragungsrate kann bei einzelnen Schreibvorgängen auf ein Band höher sein, wenn die Dateien von der Bandstation komprimiert werden, bevor sie auf das Band geschrieben werden. Auf dem System ist die Komprimierung durch die Bandstationen eingeschaltet. Die Dateien sind jedoch zum größten Teil binäre Dateien, die sich nur in sehr geringem Maße komprimieren lassen.

Durchschnittliche Dauer eines Migrationsvorgangs

Für die durchschnittliche Dauer eines Migrationsvorgangs lässt sich folgende Formel ermitteln:

$$\begin{aligned}
 t_{\text{Migrationszeit}} &= t_{\text{Mountzeit}} + t_{\text{Spulzeit}} + X_{\text{Anzahl_MB}} * Y_{\text{Migrationsdauer_für_ein_MB}} \\
 &= 35s + 58s + X_{\text{Anzahl_MB}} * \frac{1}{30}s;
 \end{aligned}$$

Migration von premigrierten Daten

Werden premigrierte Dateien migriert, die nach der Premigration nicht modifiziert wurden, so verkürzt sich die Migrationsdauer deutlich. Die Dateien müssen nicht mehr auf ein externes Speichermedium geschrieben werden, sondern sie werden lediglich auf dem lokalen Dateisystem gelöscht und durch einen 'Stub File' ersetzt. Die Migration einer nicht modifizierten premigrierten Datei dauert unabhängig von der Größe ca. 1 Sekunde.

4.4 Wie lange dauert der Recall einer Datei?

Mountzeit- und Spulzeit

Mount- und Spulzeiten sind identisch mit den Zeiten, die sich bei der Migration ergeben.

Datenübertragungsrate

Zur Bestimmung der Datenübertragungsrate werden einhundert 1 GB große Dateien sequentiell von einem gemounteten Band gelesen und auf das lokale Dateisystem geschrieben. Die Zeit jedes einzelnen Lese- und Schreibvorgangs wird gemessen. Zwischen den einzelnen Lesevorgängen wird das Band nicht neu gemountet.

Die Messungen ergaben eine Übertragungsrate von 30 MB/s. Folglich dauert es ca. 34 Sekunden, 1 GB auf das Dateisystem zu schreiben.

Durchschnittliche Dauer eines Recallvorgangs

Für die durchschnittliche Dauer eines Recallvorgangs lässt sich folgende Formel ermitteln:

$$\begin{aligned} t_{\text{Recallzeit}} &= t_{\text{Mountzeit}} + t_{\text{Spulzeit}} + X_{\text{Anzahl_MB}} * Y_{\text{Migrationsdauer_für_ein_MB}} \\ &= 35s + 58s + X_{\text{Anzahl_MB}} * \frac{1}{30}s; \end{aligned}$$

Wie die Untersuchungen zeigen, sind Migrationszeit und Recallzeit identisch.

4.5 Wie verhält sich das Datenwachstum auf dem Dateisystem?

Es wird untersucht, wie das Datenvolumen auf den drei Benutzerdateisystemen anwächst. Hierzu wird der Füllungsgrad der Dateisysteme gemessen. Der Füllungsgrad des Dateisystems ist wie folgt definiert:

$$\text{Füllungsgrad} = \frac{\sum \text{migriertes Datenvol.} + \sum \text{nicht migriertes Datenvol.}}{\text{Gesamt-speicherplatz des Dateisystems}} * 100[\%]$$

Der Füllungsgrad ist im Folgenden graphisch dargestellt. Ein Anstieg des Füllungsgrades um 1% bedeutet eine Zunahme des Datenvolumens um 10,92GB.

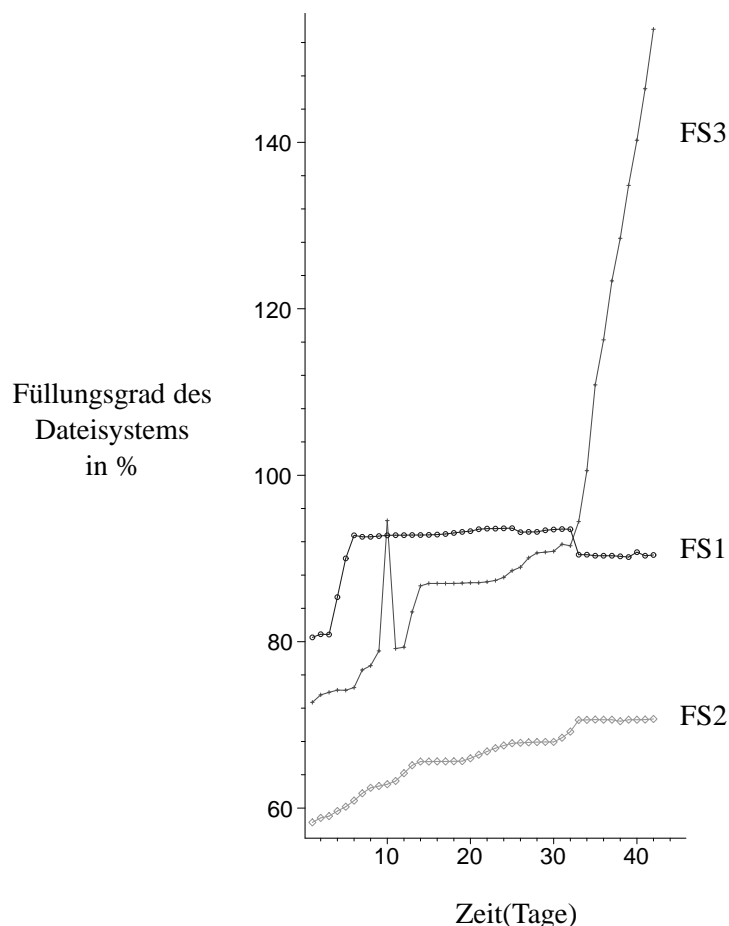


Abbildung 4.3: Datenwachstum auf Dateisystemen FS1 , FS2 , FS3

Die drei Kurven zeigen deutlich, dass das Datenvolumen auf Dateisystemen sehr unterschiedlich und unvorhersagbar ansteigen kann.

Auf dem Dateisystem FS1 kann beobachtet werden, dass das Datenvolumen auf einem Dateisystem sprunghaft ansteigen kann. Der Füllungsgrad des Dateisystems liegt zu Beginn bei ca. 80%. Innerhalb von drei Tagen steigt er auf 93% an und bleibt in den nachfolgenden Tagen relativ konstant bei diesem Wert.

Hingegen ist das Dateisystem FS2 ein Beispiel für einen kontinuierlichen Anstieg des Datenvolumens. Der Füllungsgrad des Dateisystems steigt kontinuierlich von 58% auf 70% an.

Das Dateisystem FS3 macht wiederum deutlich, wie unkalkulierbar das Datenwachstum auf einem Dateisystem ist. Der Füllungsgrad des Dateisystems steigt anfangs stetig von 72% auf 79% an. Am zehnten Tag steigt er sprunghaft auf 95% an. Am folgenden Tag sinkt er wieder um 16% und steigt dann kontinuierlich an. Ab dem 33. Tag nimmt der Füllungsgrad des Dateisystems täglich zwischen 5-10% zu. Wie der Grafik zu entnehmen ist, wird das maximal mögliche Datenvolumen auf dem lokalen Dateisystem überschritten. Dies ist möglich, da Daten zwischenzeitlich migriert wurden.

4.6 Zusammenhang zwischen Zugriffswahrscheinlichkeit und Zeit seit dem letzten Dateizugriff

4.6.1 Gibt es einen Zusammenhang zwischen der Zugriffswahrscheinlichkeit und der Zeit seit dem letzten Zugriff?

Im Folgenden wird untersucht, ob die Wahrscheinlichkeit, dass auf eine Datei zugegriffen wird, abhängig ist von der Zeit, die seit dem letzten Zugriff vergangen ist.

Um dies zu untersuchen, wird auf jedem der drei Dateisysteme über einen Zeitraum von 4 Wochen (28 Tagen) täglich protokolliert, auf Dateien welchen Alters innerhalb der letzten 24 Stunden ein Zugriff stattgefunden hat. Bei dieser Betrachtung werden nur Werktage berücksichtigt, da sich an den arbeitsfreien Tagen das Zugriffsverhalten auf Dateien merklich anders gestaltet.

An jedem Tag i ($i=1,\dots,28$) werden die Dateien entsprechend ihres Alters in eine Alterskategorie A einsortiert. Alter bedeutet in diesem Zusammenhang nicht die Dauer des Bestehens einer Datei, sondern die Dauer seit dem letzten Zugriff auf die Datei. Die verschiedenen Alterskategorien sind in Tabelle 4.1 dargestellt.

Alter in Wochen	Alterskategorie A
1 Woche	1
2 Wochen	2
3 Wochen	3
...	...
älter als 12 Wochen	13

Tabelle 4.1: Einteilung der Alterskategorien

An jedem Tag $i+1$ werden in jeder Alterskategorie A die Dateien, die am Tag zuvor (Tag i) in die Alterskategorie A fielen, betrachtet. Von diesen Dateien wird die Anzahl der Dateien, auf die innerhalb des letzten Tages zugegriffen wurde, bestimmt.

Mit Hilfe dieser Daten lässt sich für jede der 13 Alterskategorien jeweils ein Wert für die relative Häufigkeit, dass auf eine Datei zugegriffen wird, wie folgt bestimmen.

Y_i^A = Anzahl zugegriffener Dateien in Alterskategorie A am Tag i

Z_i^A = Anzahl Dateien in Alterskategorie A am Tag i

$$\text{rel. Häufigkeit(Zugriff auf Dateien aus Kategorie A)} = \frac{\sum_{i=1}^{27} Y_i^A}{\sum_{i=1}^{27} Z_i^A}$$

Aus diesen relativen Häufigkeiten wird auf die Zugriffswahrscheinlichkeiten geschlossen. Die folgenden Grafiken zeigen für jedes Dateisystem die Wahrscheinlichkeit, dass auf eine Datei aus der Alterskategorie A innerhalb von 24 Stunden zugegriffen wird.

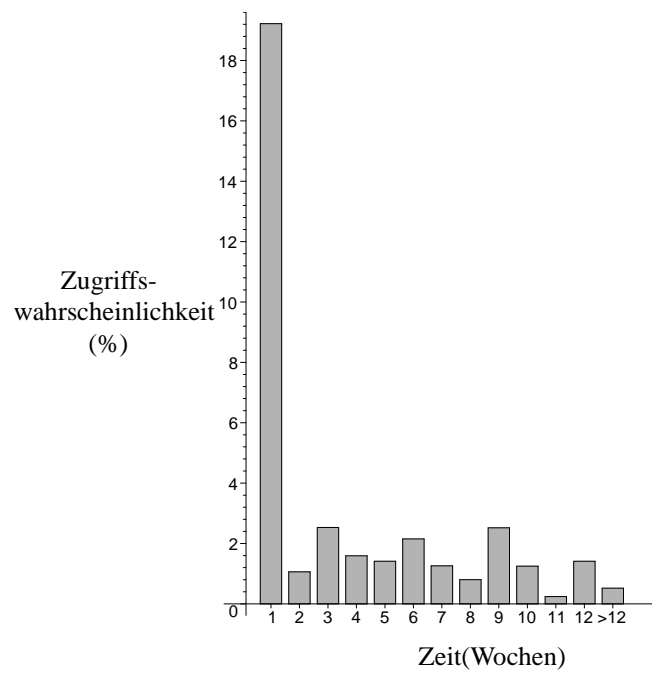


Abbildung 4.4: Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS1)

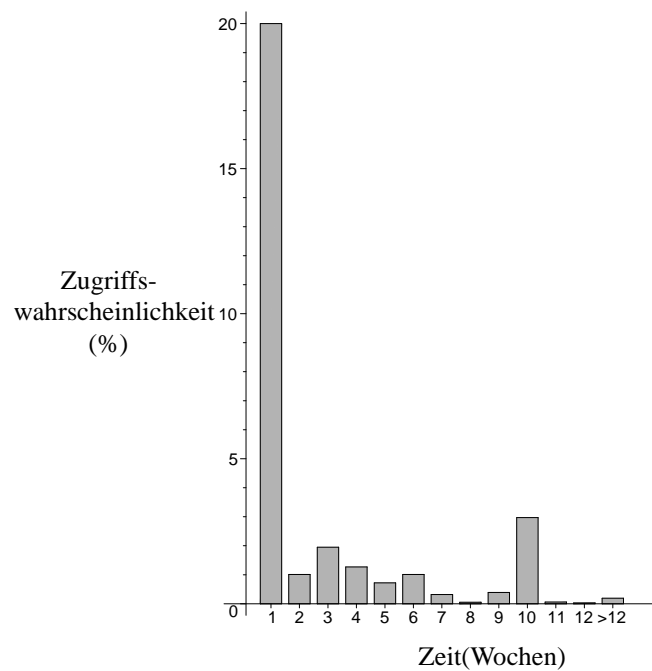


Abbildung 4.5: Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS2)

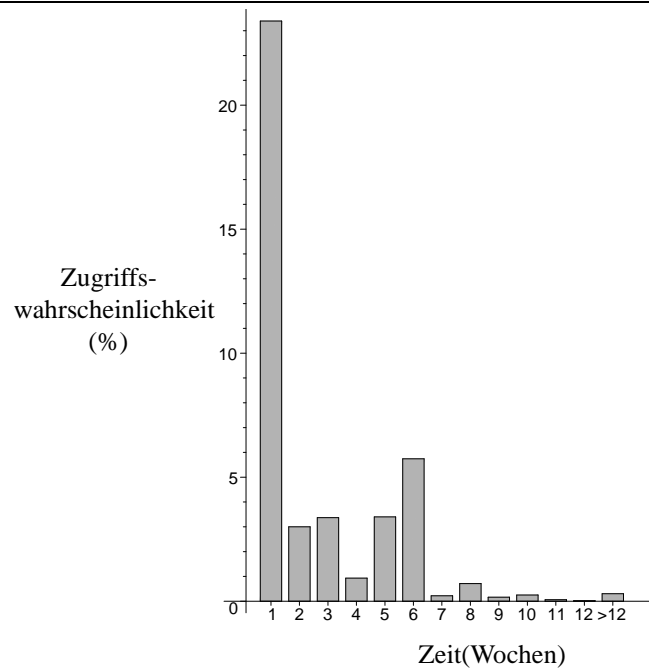


Abbildung 4.6: Zugriffswahrscheinlichkeit in Abhängigkeit vom Alter der berücksichtigten Dateien (FS3)

Die Abbildungen 4.4, 4.5 und 4.6 zeigen deutlich, dass die Wahrscheinlichkeit, dass auf eine Datei im Laufe eines Tages zugegriffen wird, bei Dateien der Alterskategorie 1 am größten ist. Sie liegt bei allen drei Dateisystemen bei mindestens 19%. Bei Dateien, die älter als eine Woche sind, liegt die Zugriffswahrscheinlichkeit immer unter 5% pro Alterskategorie, in den meisten Fällen sogar unter 1%.

4.6.2 Welchen Anteil haben Dateien der verschiedenen Alterskategorien an der Menge der täglich zugegriffenen Dateien?

Die folgende Untersuchung soll feststellen, ob es einen Zusammenhang zwischen der Alterskategorie und der Wahrscheinlichkeit, dass es sich bei einem Zugriff um eine Datei aus der Alterskategorie A handelt, gibt.

Es wird die gleiche Untersuchung wie im letzten Kapitel (4.6.1) durchgeführt jedoch mit dem Unterschied, dass dort alle Dateien einbezogen wurden, während hier nur die Dateien in die Rechnung einfließen, auf die zugegriffen worden ist.

Für jede Alterskategorie wird folgender Wert berechnet:

Y_i^A = Anzahl zugegriffener Dateien in Alterskategorie A am Tag i

Z_i = Anzahl zugegriffener Dateien auf Dateisystem am Tag i

$$\text{rel. Häufigkeit(Zugriff auf Dateien aus Kategorie A)} = \frac{\sum_{i=1}^{27} Y_i^A}{\sum_{i=1}^{27} Z_i}$$

Aus diesen relativen Häufigkeiten wird auf die Wahrscheinlichkeiten geschlossen, dass es sich bei einem Dateizugriff um eine Datei aus der Alterskategorie A handelt.

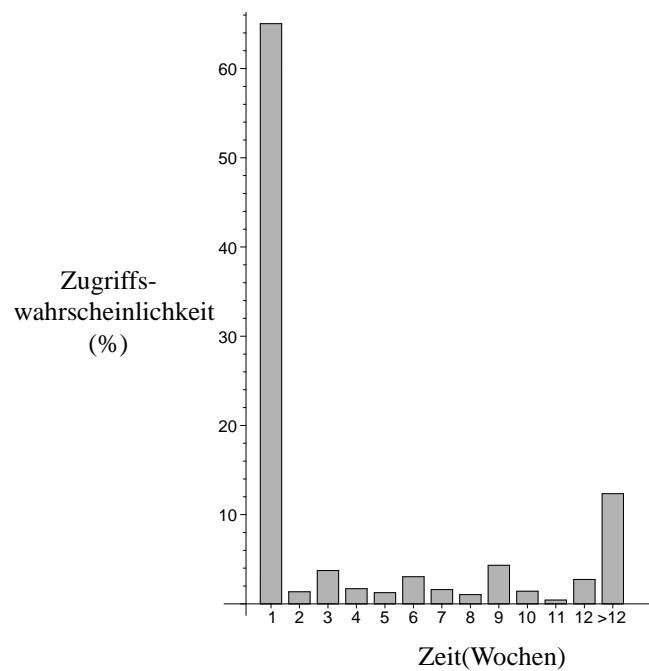


Abbildung 4.7: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS1)

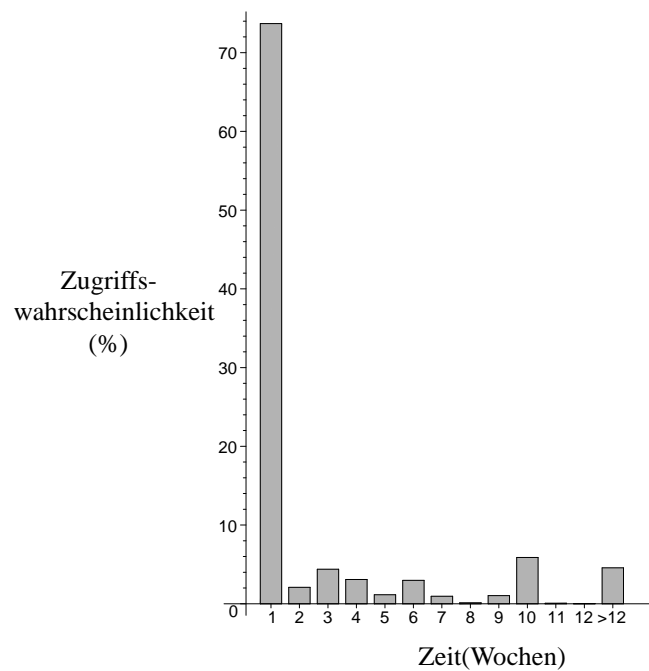


Abbildung 4.8: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS2)

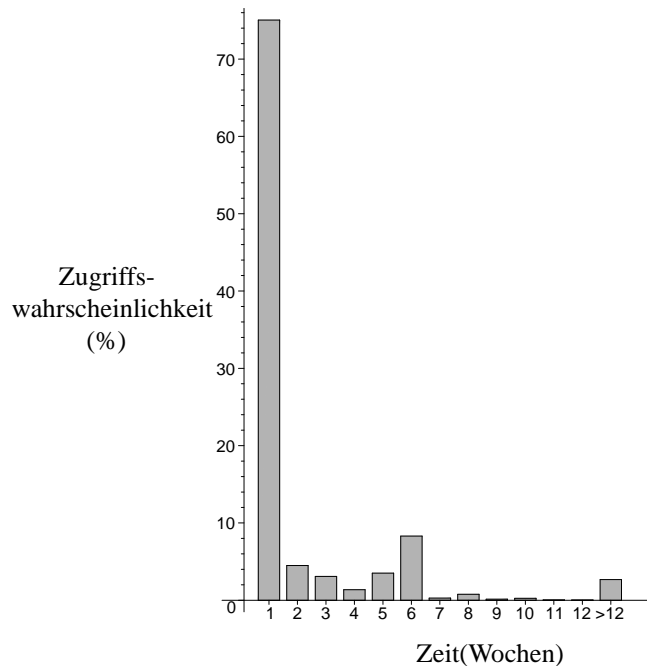


Abbildung 4.9: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Alterskategorie A handelt (FS3)

Die Auswertungen (siehe Abbildungen 4.7, 4.8 4.9) zeigen deutlich, dass es sich bei einem Dateizugriff auf diesen Dateisystemen in mindestens 65% der Fälle um eine Datei handelt, auf die in den letzten sieben Tagen zugegriffen wurde.

4.7 Zusammenhang zwischen Zugriffswahrscheinlichkeit und Dateigröße

4.7.1 Welchen Anteil haben Dateien der verschiedenen Größenkategorien an der Menge der täglich zugegriffenen Dateien?

Um die Höhe des Anteils zu bestimmen, den die verschiedenen Größenkategorien an der Summe der täglich zugegriffenen Dateien haben, wird auf jedem der drei Dateisysteme über einen Zeitraum von 4 Wochen täglich protokolliert, auf Dateien welcher Größe ein Zugriff innerhalb von 24 Stunden stattgefunden hat.

An jedem Tag i werden die Dateien entsprechend ihrer Größe in eine Größenkategorie G einsortiert. Es gibt folgende Kategorien:

Größe	Größenkategorie G
kleiner 100MB	1
100MB-1GB	2
1GB-10GB	3
größer 10GB	4

Tabelle 4.2: Einteilung der Größenkategorien

Für die verschiedenen Größenkategorien wird die Anzahl Dateien, die an dem Tag i in die entsprechende Größenkategorien G fallen, bestimmt. Am nächsten Tag (Tag $i+1$) werden in jeder Größenkategorie G die Dateien, die am Tag zuvor (Tag i) in die Größenkategorie G fielen, betrachtet. Es wird die Anzahl Dateien, auf die in den letzten 24 Stunden zugegriffen wurde, bestimmt.

Mit Hilfe dieser Daten lassen sich für die vier Größenkategorien jeweils ein Wert für die relative Häufigkeit, dass auf eine Datei zugegriffen wird, wie folgt bestimmen:

Y_i^G = Anzahl zugegriffener Dateien in Größenkategorie G am Tag i

Z_i = Anzahl zugegriffener Dateien auf Dateisystem am Tag i

$$\text{rel. Häufigkeit(Zugriff auf Dateien aus Kategorie } G) = \frac{\sum_{i=1}^{27} Y_i^G}{\sum_{i=1}^{27} Z_i}$$

Aus diesen relativen Häufigkeiten wird auf die Wahrscheinlichkeiten geschlossen, dass es sich bei einem Dateizugriff um eine Datei aus der Größenkategorie G handelt. Die folgenden Grafiken zeigen für jedes Dateisystem die Zugriffswahrscheinlichkeit auf Dateien in den einzelnen Größenkategorien.

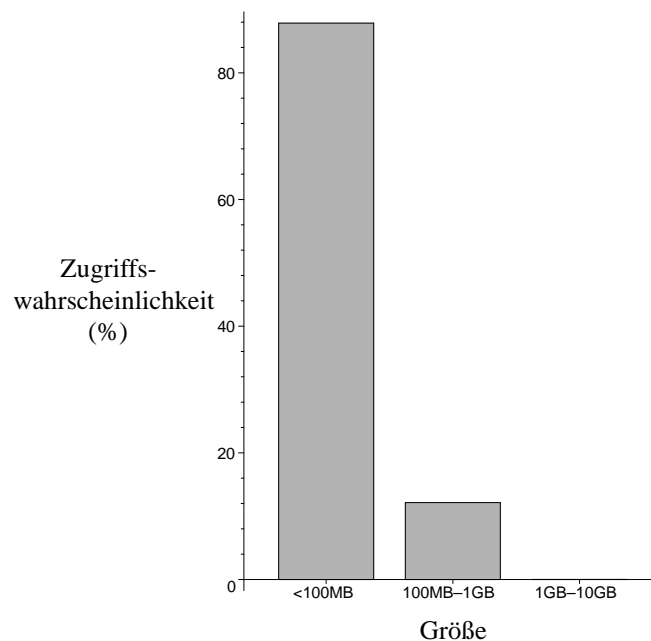


Abbildung 4.10: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS1)

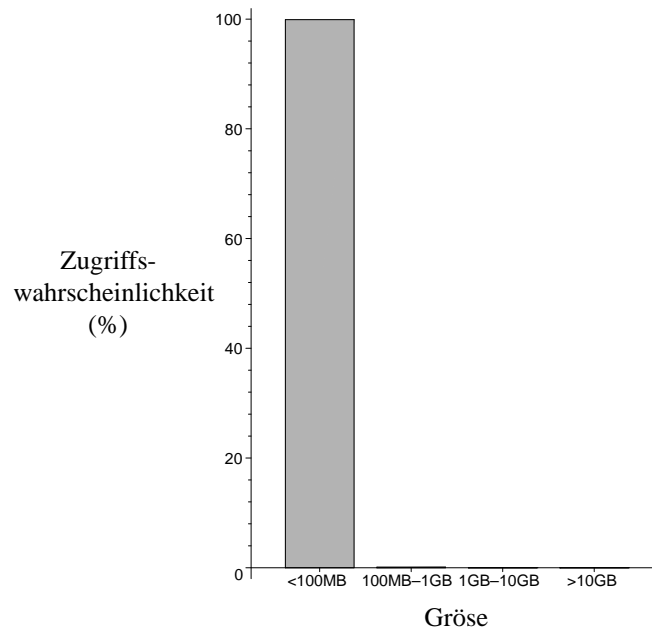


Abbildung 4.11: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS2)

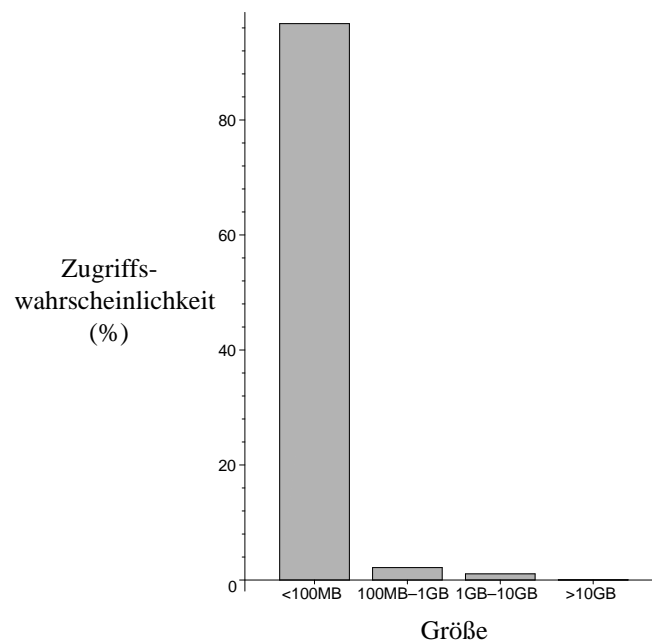


Abbildung 4.12: Wahrscheinlichkeit, dass es sich bei einem Dateizugriff um eine Datei aus Größenkategorie G handelt (FS3)

Die Abbildung 4.10 zeigt, dass es sich bei 87% der Dateizugriffe um eine Datei aus der Größenkategorie 1 handelt. In 12% der Fälle stammt die Datei aus der Größenkategorie 2. Auf Dateien, die zwischen 1-10 GB groß sind, wurde nur in 0.02% der Fälle zugegriffen. Auf Dateisystem FS2 (siehe Abbildungen 4.11) liegt die Wahrscheinlichkeit, dass es sich bei einem Dateizugriff, um eine Datei aus der Größenkategorie 1 handelt, bei nahezu 100%. Auf dem Dateisystem FS3 (siehe Abbildungen 4.12) liegt die Wahrscheinlichkeit bei 97%.

4.8 Zusammenhang zwischen der Zugriffswahrscheinlichkeit, der Dateigröße und dem letzten Zugriff

Es wird untersucht, ob es einen Zusammenhang zwischen dem Zugriff auf Dateien der verschiedenen Alterskategorien und deren Größe gibt. Hierzu werden die Dateien sowohl in Alterskategorie als auch in Größenkategorien eingeordnet. Die Größenkategorien sind die gleichen wie in Kapitel 4.7.1 (siehe Tabelle 4.2). Alle Dateien, auf die nicht zugegriffen wurde, werden wie in den Untersuchungen der vorherigen Kapitel (siehe Tabelle 4.1) einsortiert. Die restlichen Dateien bilden eine 14. Kategorie, so dass es folglich statt 13 Alterskategorien 14 modifizierte Alterskategorien gibt, die wie folgt aussehen:

Alter	Zugriff ja(+)/nein(-)	Alterskategorie A
1 Woche	-	1
2 Wochen	-	2
...
älter als 12 Wochen	-	13
älter als 0 Tage	+	14

Tabelle 4.3: Klasseneinteilung nach modifizierten Alterskategorien

Die Verteilung der Dateien eines Dateisystems auf die einzelnen modifizierten Alterskategorien und den Größenkategorien kann, wie in der Tabelle 4.4 exemplarisch für das Dateisystem FS1 gezeigt, dargestellt werden.

mod. Alters- kategorie \ Größen- kategorie	1	2	3
1	153128	28783	38
2	75811	8155	23
3	83935	11879	25
4	59875	10155	72
5	49150	9494	10
6	85908	6312	101
7	81507	1857	101
8	83963	1157	98
9	109232	1840	107
10	73481	713	159
11	117069	1327	143
12	125251	1508	101
13	1578821	4967	1683
14	58495	8084	12

Tabelle 4.4: Verteilung der Dateien des Dateisystems FS1 auf Alters- und Größenkategorien

Um die Frage zu beantworten, ob es einen Zusammenhang zwischen dem Dateizugriff in Abhängigkeit der verschiedenen Alterskategorien und deren Größe gibt, geht man von der folgenden Überlegung aus.

Bezeichne p_{A_i} die Wahrscheinlichkeit, dass auf eine Datei, die i Wochen alt ist, zugegriffen wird. Ferner sei p_{G_j} die Wahrscheinlichkeit, dass eine Datei aus der Größenkategorie j stammt. Dann müsste bei Unabhängigkeit der beiden Merkmale Alter/Zugriff und Größe einer Datei die Wahrscheinlichkeit ($p_{A_i G_j}$), dass auf eine Datei der Alterskategorie A_i und der Größenkategorie G_j zugegriffen wird, gegeben sein durch:

$$p_{A_i G_j} = p_{A_i} * p_{G_j}.$$

Zwischen den Merkmalen sind Abhängigkeiten vorhanden, wenn für mindestens eine Alters- und Größenkategorie diese Beziehung nicht erfüllt ist.

Zur Prüfung der Unabhängigkeit wird folgende Hypothese in einem Test verifiziert:

$$H_0^U : p_{A_i G_j} = p_{A_i} * p_{G_j}$$

gegen

$$H_1^U : p_{A_i G_j} \neq p_{A_i} * p_{G_j}$$

Bevor dieser Test gemacht werden kann, ist die Einführung eines loglinearen Modells notwendig, wie es in "Statistik - Lehr- und Handbuch der angewandten Statistik" von J. Hartung [3] beschrieben ist.

Nach dem loglinearen Modell lassen sich die zu den empirischen Häufigkeiten $n_{A_i G_j}$ (siehe Tab. 4.4) korrespondierenden unbekannten theoretischen Wahrscheinlichkeiten $p_{A_i G_j}$ mit den (unbekannten) Parametern u' , $u_{1(A_i)}$, $u_{2(G_j)}$ und $u_{1,2(A_i G_j)}$ in der folgenden Form darstellen:

$$p_{A_i G_j} = e^{u' + u_{1(A_i)} + u_{2(G_j)} + u_{1,2(A_i G_j)}}$$

Dabei sollen, um eine eindeutige Darstellung zu gewährleisten, die Nebenbedingungen

$$\begin{aligned} \sum_{i=1}^r u_{1(A_i)} &= \sum_{j=1}^s u_{2(G_j)} = 0 \\ \sum_{i=1}^r u_{1,2(A_i G_j)} &= \sum_{j=1}^s u_{1,2(A_i G_j)} = 0 \end{aligned}$$

erfüllt sein. r entspricht der Anzahl der modifizierten Alterskategorien und s der Anzahl der Größenkategorien.

Bezeichnet man die erwarteten Häufigkeiten mit $m_{A_i G_j}$, so erhält man:

$$m_{A_i G_j} = n \cdot p_{A_i G_j} = e^{\ln n + u' + u_{1(A_i)} + u_{2(G_j)} + u_{1,2(A_i G_j)}} = e^{u + u_{1(A_i)} + u_{2(G_j)} + u_{1,2(A_i G_j)}},$$

wobei $u = \ln n + u'$ gesetzt ist. Es werden also bis auf eine Ausnahme die gleichen Parameter wie oben verwendet.

Das führt auch bei der Aufstellung der zu testenden Hypothese zu gleichartigen Formulierungen für $p_{A_i G_j}$ und $m_{A_i G_j}$. Deshalb genügt es, eine dieser beiden Größen zu untersuchen. Das wird im Folgenden die zu erwartenden Häufigkeiten $m_{A_i G_j}$ sein.

Es gilt:

$$\ln m_{A_i G_j} = u + u_{1(A_i)} + u_{2(G_j)} + u_{1,2(A_i G_j)}$$

Für die einzelnen Parameter erhält man nun die Ausdrücke

$$\begin{aligned} u &= \frac{1}{rs} \sum_{i=1}^r \sum_{j=1}^s \ln m_{A_i G_j}, \\ u_{1(A_i)} &= \frac{1}{s} \sum_{j=1}^s \ln m_{A_i G_j} - \frac{1}{rs} \sum_{i=1}^r \sum_{j=1}^s \ln m_{A_i G_j}, \end{aligned}$$

$$u_{2(G_j)} = \frac{1}{r} \sum_{i=1}^r \ln m_{A_i G_j} - \frac{1}{rs} \sum_{i=1}^r \sum_{j=1}^s \ln m_{A_i G_j}$$

$$\text{und } u_{1,2(A_i G_j)} = \ln m_{A_i G_j} - \frac{1}{s} \sum_{j=1}^s \ln m_{A_i G_j} - \frac{1}{r} \sum_{i=1}^r \ln m_{A_i G_j} + \frac{1}{rs} \sum_{i=1}^r \sum_{j=1}^s \ln m_{A_i G_j}$$

Die schon zuvor aufgestellte Unabhängigkeitshypothese

$$H_0 : p_{A_i G_j} = p_{A_i} * p_{G_j} \quad \text{bzw.} \quad m_{A_i G_j} = \frac{m_{A_i} * m_{G_j}}{n}$$

ist äquivalent mit der Hypothese

$$H_0 : u_{1,2(A_i G_j)} = 0$$

In dem hier angewendeten Testverfahren für die Unabhängigkeitshypothese werden zunächst die erwarteten Häufigkeiten $m_{A_i G_j}$ geschätzt. Dies wird mit dem Maximum-Likelihood-Verfahren (ML-Verfahren) gemacht. Es ergibt sich als Schätzung der erwarteten Häufigkeiten

$$m_{A_i G_j} = \frac{n_{A_i} * n_{G_j}}{n}$$

Diese geschätzten erwarteten Häufigkeiten werden mit den tatsächlich beobachteten Häufigkeiten verglichen. Dies geschieht über die χ^2 Teststatistik

$$\chi^2 = \sum_{A_i=1}^r \sum_{G_j=1}^s \frac{(n_{A_i G_j} - m_{A_i G_j})^2}{m_{A_i G_j}}.$$

Diese Statistik ist unter der zu testenden Nullhypothese asymptotisch χ^2 verteilt mit den Freiheitsgraden $(r-1)(s-1)$.

Bestimmt man die erwarteten Häufigkeiten $m_{A_i G_j}$ und berechnet damit die χ^2 Teststatistik, so erhält man für die drei Dateisysteme die in Tabelle 4.5 aufgeführten Werte.

Dateisystem	χ^2 Wert
FS 1	247743
FS 2	1751
FS 3	24390

Tabelle 4.5: χ^2 Werte der einzelnen Dateisysteme

Es gilt:

Ist der berechnete Wert größer als das $(1-\alpha)$ Quantil der zugehörigen χ^2 Verteilung, so muss die Hypothese zum Niveau α verworfen werden. Das 90% Quantil der χ^2 Verteilung, hat den Wert 35. Folglich muss die Hypothese, dass Zugriffe auf Dateien der verschiedenen Alterskategorien von deren Größe unabhängig sind, aufgrund der großen berechneten Werte für alle drei Dateisysteme verworfen werden. D.h. für die Zugriffswahrscheinlichkeit gibt es einen Zusammenhang zwischen Größe und Alter.

Im Folgenden ist die Zugriffswahrscheinlichkeit auf eine Datei in Abhängigkeit von der Dateigröße und des Dateialters dargestellt. Die Einteilung in Alters- bzw. Größenkategorien sind wie in den Tabellen 4.1 und 4.2 dargelegt.

Größen- kategorie Alterskategorie	1	2	3
1	57.56	7.45	$0.11 * 10^{-1}$
2	1.35	0	0
3	2.07	1.66	0
4	1.47	0.24	0
5	1.16	$0.99 * 10^{-1}$	0
6	0.60	2.44	0
7	1.60	0	0
8	0.95	$0.81 * 10^{-1}$	0
9	4.32	0	0
10	1.39	$0.34 * 10^{-1}$	0
11	0.28	0.15	0
12	2.73	0	0
13	12.35	0	$0.57 * 10^{-2}$

Tabelle 4.6: Zugriffswahrscheinlichkeit (in %) auf Dateien des Dateisystems FS1 der verschiedenen Alters- und Größenkategorien

Anhand der in der Tabelle 4.6 dargelegten Zugriffswahrscheinlichkeiten kann eine optimale Migrationsreihenfolge bzgl. Alter und Größe einer Datei bestimmt werden, indem die Dateien entsprechend ihrer Zugriffswahrscheinlichkeit in ansteigender Reihenfolge migriert werden. Die von der Software TSM/HSM verwendete Formel zur Bestimmung der Migrationsreihenfolge ermöglicht jedoch eine so variablen Auswahl nicht und somit kann diese Information für die Optimierung der Migrationsstrategie nicht genutzt werden.

Kapitel 5

Ergebnisse

5.1 Wie kann immer freier Speicherplatz zur Abspeicherung von Dateien auf dem Dateisystem garantiert werden?

Aufgrund der Tatsache, dass die Schreibgeschwindigkeit von Daten auf das lokale Dateisystem mit 500 MB/s erheblich schneller ist als die Schreibgeschwindigkeit auf Band von 30 MB/s, ist es nicht möglich, dass Volllaufen eines Dateisystems zu verhindern. Durch das Einstellen des oberen und unteren Schwellenwertes sowie des Premigrationswertes kann lediglich dafür gesorgt werden, dass im Normalfall immer ausreichend Platz zur Abspeicherung von Daten vorhanden ist.

Diese drei Faktoren werden in Abhängigkeit von der Migrationsdauer und dem Datenwachstum eingestellt.

5.1.1 Schwellenwerte

Auf jedem Dateisystem sollte als Puffer immer mindestens so viel Speicherplatz frei sein, wie das Maximum an täglich anfallendem Datenvolumen beträgt. Um die Verfügbarkeit des freien Speicherplatzes zu erhöhen, wird der obere Schwellenwert so gesetzt, dass das Doppelte des beobachteten Maximums an täglich anfallendem Datenvolumen auf dem Dateisystem frei ist.

Der untere Schwellenwert wird so gesetzt, dass das Maximum an täglich anfallendem Datenvolumen migriert wird:

V_t^{max} := relativer Anteil des maximalen täglich anfallenden Datenvolumens
am Gesamtdatenvolumen des Dateisystems

$$\begin{aligned}\text{oberer Schwellenwert} &= 100\% - 2 * V_t^{max} \\ \text{unterer Schwellenwert} &= 100\% - 3 * V_t^{max}\end{aligned}$$

Das Maximum an täglichem Datenwachstum wurde auf dem Dateisystem FS3 beobachtet. Hier betrug der maximale gemessene Anstieg pro Tag ca 165 GB. Dies entspricht 15% des Dateisystems. Das bedeutet, dass ein Füllungsgrad von 85% nicht überschritten werden soll, damit Speicherplatz in Höhe des maximal zu erwartenden Datenvolumens in Reserve zur Verfügung steht.

Der obere Schwellenwert wird so gesetzt, dass weitere 165 GB frei bleiben. Folglich sollte der obere Schwellenwert auf 70% eingestellt werden.

Unter der Voraussetzung, dass nur ein Migrationsprozess läuft, dauert eine Migration von 165 GB ca. 1.5 Stunden.

Bei einer Migration soll die maximal zu erwartende Datenmenge, die im Normalfall an einem Tag

auf das Dateisystem kommt, ausgelagert werden. Dies sind 165 GB. Aus diesem Grund sollte der untere Schwellenwert auf 55% gesetzt werden.

5.1.2 Premigrationswert

Wie die Messungen in Kapitel 4.3 ergaben, dauert die Migration einer nicht modifizierten premigrierten Datei unabhängig von der Größe ca. 1 Sekunde. Der Migrationsvorgang einer Datei, die nicht premigriert ist, benötigt im Mittel 93 Sekunden, um das Band zu Mounten und an die Schreibposition zu spulen, zuzüglich weiteren $\frac{1}{30} * x$ Sekunden, um x MB auf ein Band zu schreiben. Die Migration premigrierter Dateien ist somit erheblich schneller als von Dateien, die sich nur auf dem lokalen Dateisystem befinden. Die Migration kann folglich beschleunigt werden, indem möglichst viele Dateien premigriert werden. Es ist sinnvoll, alle Dateien, die die Migrationskriterien erfüllen, nach einem Migrationsvorgang zu premigrieren. Bis zum nächsten Migrationsvorgang ist ein Teil der premigrierten Dateien allerdings modifiziert worden, so dass dieser Teil ungültig ist. Reicht das Datenvolumen der gültigen premigrierten Dateien nicht aus, so müssen weitere Dateien des Dateisystems, die den Migrationskriterien entsprechen, migriert werden.

Durch die Premigration werden vermehrt Rechnerressourcen in Form von Bandspeicherplatz und Prozessorleistung in Anspruch genommen. Da der vorhandene Speicherplatz auf Bänder derzeit 585 TB umfasst und durch Hinzufügen von weiteren Bändern nahezu verdoppelt werden kann und für I/O Aufgaben eine eigene Partition mit 8 Prozessoren zur Verfügung steht, deren Auslastung gering ist, existieren keine Engpässe in den Rechnerressourcen. Es gibt allerdings einen Nachteil in der Performance des Dateisystems, wenn Daten gleichzeitig premigriert und von dem Dateisystem gelesen bzw. geschrieben werden.

Ein weiterer Aspekt spricht für die Premigration aller zur Verfügung stehenden Dateien. Sollte ein Dateisystem oder Teile eines Dateisystems defekt sein, so müssen nicht alle Dateien aus dem Backup auf das Dateisystem kopiert werden. Premigrierte Dateien, die nicht verändert wurden, gehen in den Status migriert über. Hierzu müssen lediglich die Inodes wieder angelegt werden und die *Stub Files* wieder hergestellt werden. Das Dateisystem steht schneller wieder zur Verfügung. Die migrierten Dateien werden erst beim nächsten Dateizugriff zurückgeholt.

5.2 Wie können die Zugriffszeiten für die Benutzer minimiert werden?

Die Zeit, die beim Zurückholen einer Datei für Mount- und Spulvorgänge sowie für die Datenübertragung vergeht, kann nicht verkürzt werden. Es kann lediglich die Summe aller Wartezeiten minimiert werden, indem zum einen möglichst wenige Dateien, d.h. möglichst große Dateien, und zum anderen die Dateien migriert werden, auf die am seltensten zugegriffen wird.

5.2.1 Kann eine Migrationsstrategie bzgl. des Alters einer Datei entwickelt werden?

Wie die Untersuchungen in Kapitel 4.6.1 zeigen, ist die Wahrscheinlichkeit, dass in Zukunft auf eine Datei zugegriffen wird, bei Dateien, auf die innerhalb der letzten Woche zugegriffen wurde, deutlich höher als bei älteren Dateien. Die Zugriffswahrscheinlichkeit (siehe Abbildungen 4.4, 4.5 und 4.6) liegt bei mindestens 19 Prozent. Bei Dateien, auf die länger als eine Woche nicht zugegriffen wurde, liegt die Zugriffswahrscheinlichkeit bei höchstens 5%. Meistens ist die Wahrscheinlichkeit jedoch bedeutend niedriger.

Die Untersuchungen in Kapitel 4.6.2 lassen erkennen, dass in erster Linie auf Dateien, bei denen der letzte Dateizugriff kürzere Zeit zurückliegt, wieder zugegriffen wird (siehe Abbildungen 4.7, 4.8 und 4.9). In mindestens 65% der Fälle wird auf eine Datei zugegriffen, bei der der letzte Zugriff weniger als sieben Tage zurückliegt.

Aufgrund der hohen Zugriffswahrscheinlichkeit sollten Dateien, auf die innerhalb der letzten Woche zugegriffen wurde, nicht migriert werden. Darüber hinaus sollte bei der Migration mit den Dateien, auf die die längste Zeit nicht zugegriffen wurde, begonnen werden.

5.2.2 Kann eine Migrationsstrategie bzgl. der Größe einer Datei entwickelt werden?

Die Ergebnisse in Kapitel 4.7.1 zeigen, dass es sich in über 87% der Dateizugriffe um Dateien handelt, die kleiner als 100 MB sind (siehe Abbildungen 4.10, 4.11 und 4.12). Je größer die Dateien sind, desto geringer ist die Wahrscheinlichkeit, dass auf sie zugegriffen wird. Aus diesem Grund ist es sinnvoll, bei der Migration mit den großen Dateien zu beginnen und dann mit abnehmender Größe fortzufahren.

Es gibt darüber hinaus noch weitere Gründe, die dieses Vorgehen bekräftigen. Zum einen müssen weniger Dateien migriert werden, um eine bestimmte Menge an freiem Speicherplatz zu erhalten. Es sinkt die Wahrscheinlichkeit, dass es sich um eine migrierte Datei handelt, wenn auf eine Datei zugegriffen wird.

Zum anderen fällt die Zeit von 93 Sekunden, die bei jedem Recallvorgang für das Mounten und Spulen des Bandes vergeht, bei größeren Dateien weniger stark ins Gewicht als bei kleineren Dateien.

Für den Recallvorgang wurde in Kapitel 4.4 eine Formel bestimmt, die in Abhängigkeit von der Größe einer Datei die erwartete Recallzeit angibt:

$$t_{\text{Recallzeit}} = 35s + 58s + X_{\text{Anzahl_MB}} * \frac{1}{30}s;$$

In Abbildung 5.1 ist die Funktion graphisch dargestellt. Es ist zu erkennen, dass die Recall-Zeit direkt proportional zur Dateigröße ist. (Linearer Zusammenhang)

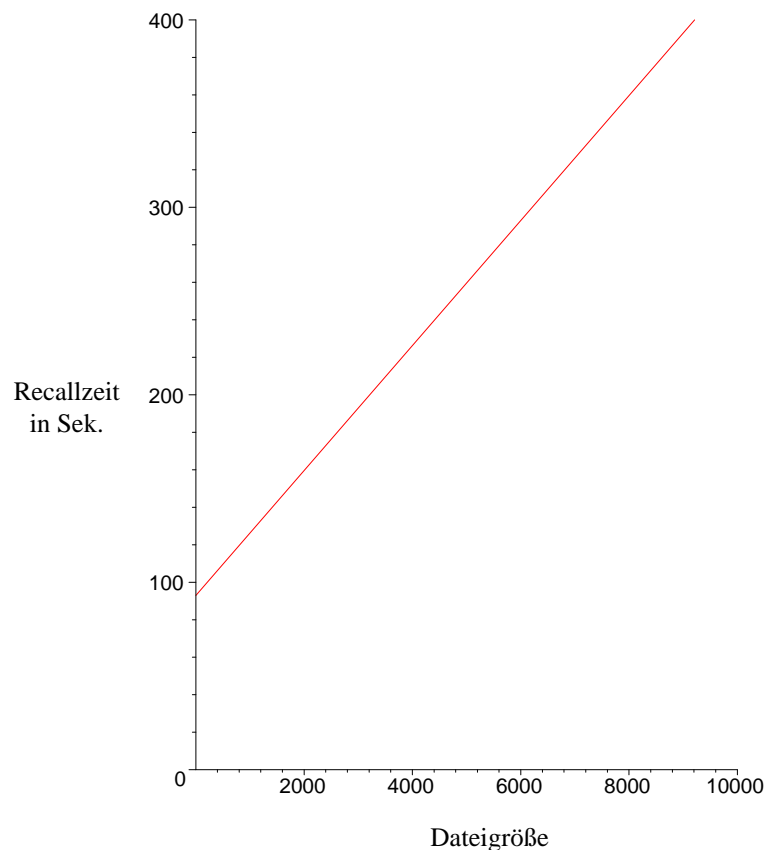


Abbildung 5.1: Recalldauer von Dateien

Aus der Formel für die durchschnittliche Dauer eines Recallvorgangs lässt sich die folgende Formel herleiten, mit deren Hilfe sich die durchschnittliche Dauer für das Zurückholen von 100 MB in Abhängigkeit von der Größe der zurückgeholten Datei berechnen lässt.

$$t_{\text{Recallzeit}_100\text{MB}} = \left(\frac{93}{X_{\text{Anzahl_MB_der_Datei}}} s + \frac{1}{30} s \right) * 100;$$

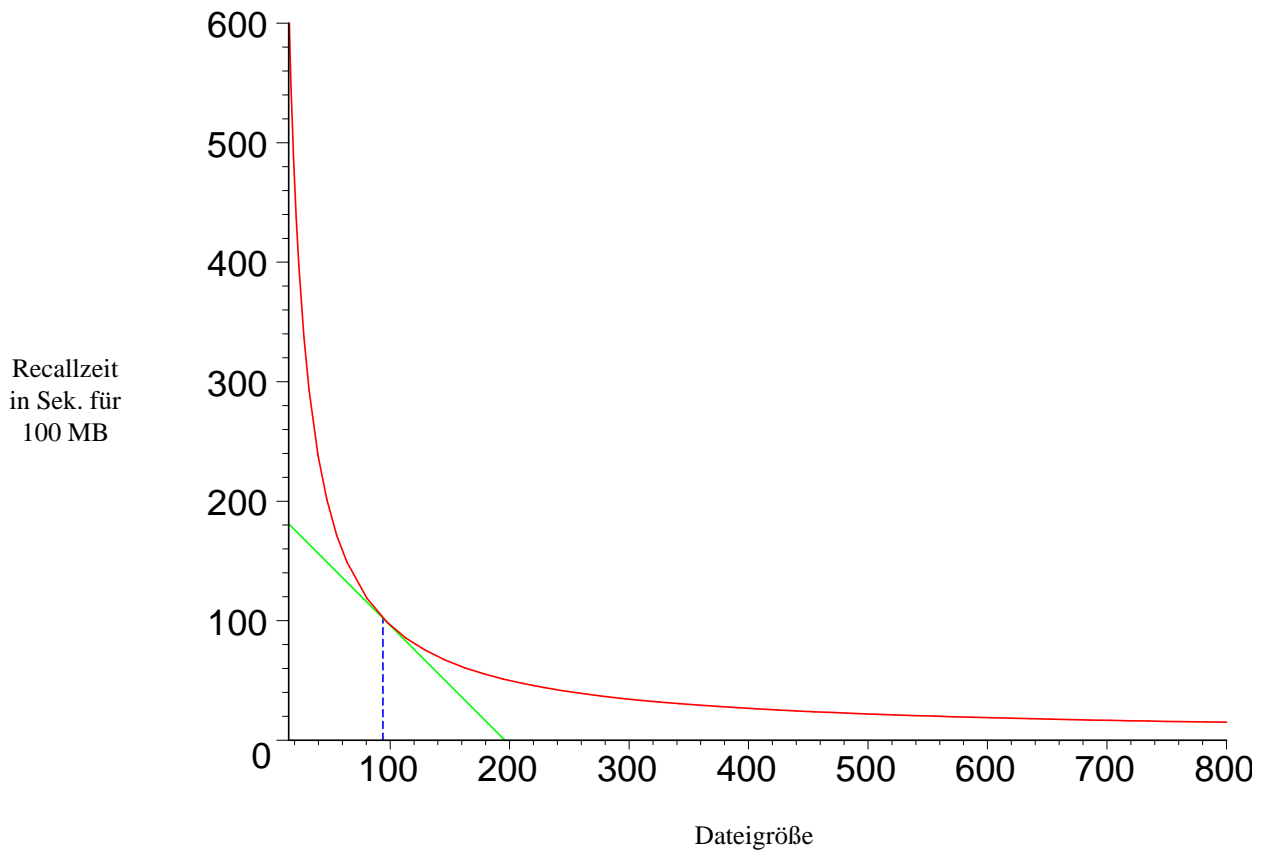


Abbildung 5.2: Durchschnittliche Dauer des Recallvorgangs für 100 MB in Abhängigkeit von der Dateigröße

Die Abbildung 5.2 zeigt die durchschnittliche Dauer für das Zurückholen von 100 MB in Abhängigkeit von der Größe der zurückgeholten Datei.

Es ist zu erkennen, dass die durchschnittliche Recall-Zeit pro 100 MB bei kleinen Dateien wesentlich länger und damit uneffizienter ist als bei großen Dateien. Bei einer Dateigröße über 100 MB liegt die durchschnittliche Recallzeit pro 100 MB unter 100 Sekunden. Ab einer Dateigröße von 96 MB verkürzt sich die Recall-Zeit für 100 MB Datenvolumen um weniger als 1 Sekunde pro zusätzliches MB ($\frac{\Delta \text{Recallzeit}}{\Delta \text{Dateigröße}} < -1$). Daher ist es sinnvoll, Dateien ab einer Größe von 100 MB zu migrieren.

5.2.3 Einfluss der gewählten Migrationsstrategie auf die Kosten

Es soll nun gezeigt werden, welchen Einfluss die Einschränkung durch Mindestalter und Mindestgröße einer Datei sowohl auf die Menge des Datenvolumens hat, das migriert werden kann, als auch auf die täglichen Wartezeiten und den damit verbundenen Kosten, die entstehen, wenn auf eine migrierte Datei zugegriffen wird.

Am Ende jeden Monats berechnet ein Abrechnungsverfahren für alle Benutzer des IBM-Supercomputers den Verbrauch der verschiedenen Betriebsmittel in Verrechnungseinheiten (VE). Der Verbrauch wird nach Belegzeiten der zugeteilten Prozessoren und der verbrauchten CPU-Zeit wie folgt berechnet:

$$\text{Verbrauch (VE)} = fc * \max((NP * TIME), CPU);$$

mit folgenden Betriebsmittelgrößen:

NP: Anzahl der zugeteilten Prozessoren

TIME: Belegzeit der Prozessoren durch den Job (h)

CPU: CPU-Zeit (h), berechnet als Summe aus der User-CPU und System-CPU-Zeit

und dem Faktor: fc: 2.1 (VE/h)

Aus dem ermittelten Verbrauch ergibt sich der Preis für die in Anspruch genommenen Betriebsmittel. Benutzer müssen pro Verrechnungseinheit einen Euro zahlen.

Zur Vereinfachung wird im Folgenden davon ausgegangen, dass der Verbrauch der Verrechnungseinheiten durch das Produkt aus der Anzahl zugeteilter Prozessoren und deren Belegzeit bestimmt ist:

$$\text{Verbrauch (VE)} = fc * NP * TIME; \quad (*)$$

Um einen Richtwert für die täglich anfallenden Kosten zu haben, die beim Zugriff auf eine migrierte Datei entstehen, wird davon ausgegangen, dass in einem Job, der auf einem Knoten läuft (32 Prozessoren), auf eine migrierte Datei zugegriffen wird und dieser Job erst weiter abgearbeitet werden kann, wenn sich die Datei vollständig auf dem lokalen Dateisystem befindet. Die hierbei entstehenden Wartezeiten erhöhen somit die Kosten für einen Job und müssen von den einzelnen Benutzern getragen werden.

Darüber hinaus wird vorhandene CPU-Leistung nicht genutzt, wenn ein Job seine reservierten Prozessoren nicht verwenden kann, weil er auf Eingabedaten wartet.

1. Migrationskriterium: Mindestdateialter

Es soll zunächst an dem Dateisystem FS1, exemplarisch für alle drei Dateisysteme, gezeigt werden, welche Auswirkung die Forderung eines Mindestdateialters hat.

Hierzu zeigt die Tabelle 5.1 für das Dateisystem FS1 in Abhängigkeit des Mindestdateialters, die folgenden Werte:

- Wahrscheinlichkeit, dass ein Benutzer auf eine migrierte Datei zugreift (in Prozent)
- Anzahl zugegriffener Dateien pro Tag, die migriert sind
- relativer Anteil des migrierten Dateivolumens
- Übertragungsdauer in Stunden pro Tag für Recall
- tägliche Kosten für Recall der gesamten zugegriffenen migrierten Dateien

Mindestalter der mig. Dateien in Wochen	P-Zugriff auf mig. Datei	Anzahl Recall pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0	100	2466	100.00	64.22	4315.46
1	34.97	862	79.37	22.48	1510.58
2	33.61	829	70.96	21.62	1452.54
3	29.88	737	61.85	19.18	1289.19
4	28.18	695	53.60	18.09	1215.55
5	26.92	664	46.84	17.27	1160.54
6	23.88	589	40.52	15.26	1025.27
7	22.27	549	36.93	14.23	956.59
8	21.24	524	33.81	13.57	911.96
9	16.92	417	29.57	10.82	727.00
10	15.51	383	27.63	9.92	666.29
11	15.08	372	23.59	9.63	647.01
12	12.35	305	19.63	7.88	529.24

Tabelle 5.1: Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS1

Wären alle Dateien des lokalen Dateisystems migriert worden, so wären täglich ca. 2500 Dateien auf das Dateisystem zurückgeholt worden. Dies hätte nach der vereinfachten Kostenformel (*) Ausgaben in Höhe von ca. 4300 Euro verursacht. Bei Einschränkung der Migration auf Dateien, die älter als eine Woche sind, wäre die Wahrscheinlichkeit, dass auf eine migrierte Datei zugegriffen wird, von 100% auf 34.97% gesunken. Dies hätte zur Folge gehabt, dass nur noch ca. 850 migrierte Dateien zurückgeholt worden wären und somit die Kosten auf ca. 1500 Euro gesunken wären. Sowohl die Anzahl der migrierten zugegriffenen Dateien als auch die Kosten wären um ca. $\frac{3}{4}$ niedriger. Es wären ca. 80% des Dateivolumens migriert. Hätte man das Mindestdateialter höher angesetzt, dann hätten bei geringerer Kostenersparnis erheblich weniger Dateien migriert werden können.

2. Migrationskriterium: Mindestdateigröße

Im Folgenden wird anhand des Dateisystems FS1 (als Beispiel für alle drei Dateisysteme) dargelegt, welche Konsequenzen die Forderung einer Mindestgröße hat. Hierzu werden die gleichen Aspekte, die zuvor in Abhängigkeit vom Mindestdateialter betrachtet wurden, nun in Abhängigkeit von der Mindestdateigröße in der Tabelle 5.2 dargestellt. Es wird allerdings davon ausgegangen, dass nur Dateien, die ein Mindestalter von 1 Woche haben, migriert werden.

Mindestgröße der mig. Dateien	P-Zugriff auf mig. Datei	Anzahl Recall pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0 MB	34.97	862	79.37	22.48	1510.58
100 MB	4.69	116	74.08	3.17	212.85
1 GB	0.01	0	15.87	0.01	0.57
10 GB	0.00	0	0.00	0.00	0.00

Tabelle 5.2: Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS1

Fordert man eine Mindestdateigröße von 100 MB, so werden noch ca. 75% des Datenvolumens migriert und die Kosten sinken auf ca. 200 Euro. Würde man eine Mindestdateigröße von 1 GB fordern, dann würden zwar fast keine Kosten entstehen, aber es könnten nur knappe 16% des gesamten Dateivolumens migriert werden. Dies wäre nicht erstrebenswert, da auf der ersten Speicherebene wesentlich weniger Speicherkapazität zur Verfügung steht als auf der zweiten Ebene.

Auf den anderen beiden Dateisystemen sind die Auswirkungen aufgrund der Festlegung eines Mindestdateialters und einer Mindestdateigröße ähnlich. Im Anhang A sind hierzu weitere Tabellen zu finden.

5.2.4 Ergebnis: Migrationsstrategie

Der obere und untere Schwellenwert eines Dateisystems wird in Abhängigkeit vom beobachteten täglichen Datenwachstum auf dem Dateisystem eingestellt. Der obere Schwellenwert ist so einzustellen, dass eine Migration einsetzt, wenn weniger als das Doppelte des beobachteten Maximums an täglich anfallendem Datenvolumen auf dem Dateisystem frei ist. Der untere Schwellenwert wird so gesetzt, dass das Maximum an täglich anfallendem Datenvolumen migriert wird.

Das Mindestalter einer Datei, das gefordert wird, damit diese Datei migriert werden kann, hängt von der Wahrscheinlichkeit ab, dass auf eine Datei dieses Alters zugegriffen wird.

Ebenso hängt die Mindestgröße einer Datei, die gefordert wird, damit sie migriert werden kann, von der Zugriffswahrscheinlichkeit für Dateien dieser Größe ab. Darüber hinaus sollte die Mindestgröße einer Datei mindestens 100 MB betragen, da ein Recall von kleineren Dateien äußerst uneffizient ist.

Bei den gegebenen Dateigrößen und dem beobachteten Wachstum ist es sinnvoll, den oberen Schwellenwert auf 70% und den unteren Schwellenwert auf 55% einzustellen. Es sollten nur Dateien migriert werden, die älter als eine Woche und größer als 100 MB sind. Die Migrationsreihenfolge sollte sich in erster Linie nach dem Alter und in zweiter Linie nach der Dateigröße richten. Dies bedeutet, dass Dateien in abfallender Altersreihenfolge migriert werden. Gibt es Dateien gleichen Alters, so wird die größte Datei zuerst migriert und in abfallender Größenreihenfolge fortgefahren.

Kapitel 6

Ausblick

6.1 Übertragung der Migrationsstrategie auf den IBM Supercomputer Jump

Seit dem 16. Februar 2004 ist der IBM Supercomputer Jump des Forschungszentrum Jülich mit 41 Knoten und einer Peakleistung von 8.9 TFLOPS in Betrieb. Für die Benutzerdaten stehen sieben HOME-Filesystemen zur Verfügung, die jeweils eine Grösse von ca. 4.2 TB haben. Da die Dateisysteme ca. 4 Mal so groß sind wie die des untersuchten Rechners, das Datenwachstum allerdings auf beiden Systemen 165 GB pro Tag beträgt, ist der obere Schwellenwert auf 90% gesetzt. Der untere Schwellenwert ist auf 85% eingestellt.

Unabhängig von der Schwellenwertmigration wird jeden Abend eine automatische Migration eingeleitet. Dabei wird ein Skript gestartet, dass einen Migrationsvorgang auslöst, wenn der untere Schwellenwert überschritten ist, auch wenn der obere Schwellenwert noch nicht erreicht wurde. Es werden solange Dateien migriert, bis der untere Schwellenwert erreicht ist.

Nach jedem Migrationsvorgang werden alle Dateien, die die Migrationskriterien erfüllen, premigriert. Eine Datei erfüllt die Migrationskriterien, wenn

- der letzte Zugriff länger als eine Woche zurückliegt,
- die Datei eine Mindestgrösse von 100 MB hat und
- eine Backup Kopie von der Datei existiert.

Die Migrationsreihenfolge richtet sich nach dem Alter der Datei.

In der Tabelle 6.1 sind für alle sieben Dateisysteme der Füllungsgrad des Dateisystems, das Datenvolumen auf dem Dateisystem, das migrierte Datenvolumen und die Anzahl migrierter Dateien gezeigt. Hierbei ist der Füllungsgrad folgendermaßen definiert:

$$\text{Füllungsgrad} = \frac{\text{reales Datenvolumen auf Dateisystem}}{\text{Gesamtspeicherplatz des Dateisystems}} * 100[\%]$$

Die Tabelle zeigt, dass am 24.5.2004 23.343 Dateien mit einem Gesamtvolumen von ca. 13 TB von drei verschiedenen Dateisystemen migriert sind. Von diesen Dateien werden täglich ca. 85 Dateien zurückgeholt. Die Zeit, die im Durchschnitt für einen Recall vergeht, beträgt 68 Sekunden.

Durch die tägliche automatische Migration wird der obere Schwellenwert von 90% nicht erreicht.

Dateisystem	Füllungs- grad	Datenvolumen auf Dateisystem in TB	mig. Daten- volumen in TB	Anzahl mig. Dateien
FS1	84%	3.32	2.52	3704
FS2	86%	3.4	1.15	3937
FS3	37%	1.55	0	0
FS4	59%	2.48	0	0
FS5	86%	3.4	8.89	15702
FS6	1%	0.04	0	0
FS7	1%	0.04	0	0

Tabelle 6.1: Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf den Dateisystem FS1–FS7

6.2 Lebensdaueranalyse

Im Rahmen dieser Diplomarbeit werden nur binäre Auswertungen vorgenommen. Hierzu wird jede Datei des Dateisystems täglich anhand ihres Alters und ihrer Grösse in eine Kategorie eingeordnet und es wird protokolliert, ob auf sie ein Zugriff innerhalb von 24 Stunden stattgefunden hat. Aufgrund dieser Beobachtungen wird auf die Zugriffswahrscheinlichkeit innerhalb von 24 Stunden von Dateien der einzelnen Kategorien geschlossen. Sowohl diese Zugriffswahrscheinlichkeiten als auch die Effizienz des Recallprozesses, die von der Dateigrösse abhängt, bestimmen das Mindestalter und die Mindestgrösse einer zu migrierenden Datei. Bei dieser Vorgehensweise wird nicht berücksichtigt, ob es sich um einen einmaligen Dateizugriff handelt oder ob in verschiedenen Zeitintervallen auf diese Datei immer wieder zugegriffen wird. Auch bleibt das Zeitintervall zwischen zwei Zugriffen unberücksichtigt. Es gehen folglich Informationen verloren. Darüber hinaus wurden in dieser Untersuchung nur die Werktage berücksichtigt, da an den anderen Tagen das Zugriffsverhalten der Benutzer merklich anders war.

Es gibt eine weitergehende Möglichkeit die Auslagerungsstrategie von Dateien zu optimieren. Hierzu kann auf Verfahren der Lebensdaueranalyse zurückgegriffen werden. Es wird über einen gewissen Zeitraum das Zugriffsverhalten auf jede einzelnen Datei des Dateisystems beobachtet. Es wird protokolliert, ob auf eine Datei im Beobachtungszeitraum zugegriffen wurde und gegebenenfalls wie lange es bis zum Zugriff gedauert hat. Ebenfalls wird zu Beginn festgehalten wie alt und gross jede einzelne Datei ist. Diese Daten werden gesammelt und ausgewertet. Durch die Auswertung dieser detaillierten Daten, lässt sich eine optimalere Migrationsstrategie entwickeln, die nicht nur unter dem Aspekt "Heute-Morgen" optimal ist, sondern über einen längeren Zeitraum die Auslagerung optimiert. Wahrscheinlich fordert die auf diesem Wege gewonnene Migrationsstrategie jedoch eine flexiblere Auslagerung als dies mit der jetzigen zur Verfügung stehenden Software möglich ist.

Anhang A

Tabellen

Tabellen zu Kapitel 5.2.3 "Einfluss der gewählten Migrationsstrategie auf die Kosten".

A.1 1. Migrationsstrategie: Mindestdateialter

Mindestalter der mig. Dateien in Wochen	P-Zugriff auf mig. Datei	Anzahl Recall Dateien pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0	100	4005	100.00	103.53	6957.05
1	26.30	1054	96.04	27.23	1830.13
2	24.21	970	92.46	25.07	1684.50
3	19.83	794	87.91	20.53	1379.68
4	16.75	671	82.73	17.34	1165.40
5	15.60	625	79.16	16.15	1085.39
6	12.63	506	72.77	13.08	878.83
7	11.68	468	66.50	12.09	812.66
8	11.56	463	60.72	11.97	804.09
9	10.53	422	55.32	10.90	732.36
10	4.65	186	51.46	4.82	323.69
11	4.59	184	46.90	4.76	319.76
12	4.57	183	46.33	4.73	317.76

Tabelle A.1: Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS2

Mindestalter der mig. Dateien in Wochen	P-Zugriff auf mig. Datei	Anzahl Recall pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0	100	3759	100.00	98.28	6604.46
1	24.93	937	67.90	24.36	1637.27
2	20.43	768	55.57	19.91	1338.07
3	17.35	652	48.29	16.90	1135.61
4	15.98	601	45.62	15.56	1045.32
5	12.46	469	42.25	12.13	814.84
6	4.17	157	38.58	4.07	273.43
7	3.89	146	33.82	3.80	255.15
8	3.12	118	30.70	3.04	204.60
9	2.99	112	26.51	2.91	195.66
10	2.74	103	23.59	2.67	179.58
11	2.70	102	21.07	2.64	177.20
12	2.68	101	18.33	2.61	175.34

Tabelle A.2: Verschiedene Aspekte in Abhängigkeit des Mindestdateialters auf Dateisystem FS3

A.2 2. Migrationsstrategie: Mindestdateigröße

Mindestgröße der mig. Dateien	P-Zugriff auf mig. Datei	Anzahl Recall Dateien pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0 MB	26.30	1054	96.04	27.23	1830.13
100 MB	0.03	1	60.41	0.04	2.55
1 GB	0.00	0	38.19	0.00	0.00
10 GB	0.00	0	31.04	0.00	0.00

Tabelle A.3: Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS2

Mindestgröße der mig. Dateien	P-Zugriff auf mig. Datei	Anzahl Recall Dateien pro Tag	Anteil mig. Dateien am Gesamtvoll	Wartezeit für Recall in Std. pro Tag	tägl. Kosten für Recall
0 MB	24.93	937	67.90	24.36	1637.27
100 MB	0.73	28	58.35	0.83	55.88
1 GB	0.08	3	12.59	0.15	10.25
10 GB	0.00	0	0.00	0.00	0.00

Tabelle A.4: Verschiedene Aspekte in Abhängigkeit der Mindestdateigröße auf Dateisystem FS3

Literaturverzeichnis

- [1] IBM Tivoli Storage Manager - Enterprise Edition - Using the Hierarchicak Storage Management Clients - Version 5 Release 2
- [2] Cray Data Migration Facility (DMF) Administrator's Guide, publication SG2135, supports the 2.5 release of DMF
- [3] Statistik - Lehr- und Handbuch der angewandten Statistik - 10. Auflage - Hartung

